

## Caça níqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça níqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça níqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinow.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76001 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76199 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQ, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAI, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAI\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha - se quiser - outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

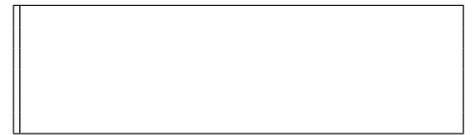
3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76018 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha - se quiser - outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
[2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76025 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76032 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha - se quiser - outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76049 - gar a

## Caça níqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça níqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça níqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76056 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinow.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamos.gif"  
onMouseDown="this.src='ima/ap. .nos_p.gif';"  
onMouseUp="this.src='ima/apostamos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76063 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap. .nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76087 - gar a

## Caça níqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça níqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça níqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamos.gif"  
onMouseDown="this.src='ima/ap. .nos_p.gif';"  
onMouseUp="this.src='ima/apostamos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76106 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinow.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap. .nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76113 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
[2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76120 - gar a

## Caça níqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça níqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça níqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinow.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76137 - gar a

## Caça níqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça níqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça níqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinow.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76144 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha - se quiser - outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinow.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamemos.gif"  
onMouseDown="this.src='ima/ap...nos_p.gif';"  
onMouseUp="this.src='ima/apostamemos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76168 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQ, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamos.gif"  
onMouseDown="this.src='ima/ap. .nos_p.gif';"  
onMouseUp="this.src='ima/apostamos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76175 - gar a

## Caça niqueis HTML+CSS+JS

O objetivo deste exercício é desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é mais ou menos trivial, mas sua conclusão com sucesso vai permitir consolidar muitas habilidades.

### O que é um caça niqueis

Uma máquina muito frequente em cassinos (físicos) e mesmo em aplicativos da web. Na sua versão física, você coloca uma moeda na máquina e abaixa uma alavanca. Com isso, diversos rolos começam a girar e param de alguma maneira (pseudo) aleatória. Quando o rolo para, ele exibe um dos símbolos que o compõe. Tais símbolos levam a uma tabela de premiação. Num aplicativo WEB (como este que estamos criando) todas estas atividades tem que ser simuladas, o que, como veremos não é difícil.

### Símbolos

Escolha 4 símbolos a seu critério. O modelo que vamos examinar usa símbolos dos times do Paraná, a saber: Coritiba, Atlético, Paraná e Operário Ferroviário de Ponta Grossa. Mas, é uma opção do desenvolvedor: pode ser banana, maçã, abacaxi e morango, pode ser sol, neve, chuva e noite, pode ser cachorro, gato, serpente e águia, pode ser o que você quiser. Feita a escolha, você precisa imagens de boa qualidade e de tamanho  $90 \times 90$  pixels. Abra um diretório (pasta) no seu pen drive de nome CACANIQU, e dentro dele abra um subdiretório (pasta) de nome IMA e coloque as 4 imagens lá.

Desenhe 3 botões de nomes APOSTAMAIS, APOSTAMENOS e JOGAR. Para dar um visual legal, crie outros botões APOSTAMAIS\_P, APOSTAMENOS\_P e JOGAR\_P com as mesmas dimensões e cores invertidas. A idéia é simbolizar para o operador que qualquer um desses botões foi pressionado (faz-se isso, trocando a imagem do botão por alguns milissegundos).

### CSS

Aproveite o CSS disponibilizado com o nome de STYLE.CSS e estude bem as definições que há nele.

### HTML de nome TABELA.HTML

É só um texto (eventualmente acompanhado de artes) indicando qual a tabela de premiação para o caça niqueis. Minha proposta de pontuação, (mas fique à vontade para criar sua própria tabela de pontuações):

2x Se os 4 símbolos forem diferentes. Como são 4 rolos, teremos aqui 24 combinações que geram este premio. São elas 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312 e 4321.

5x Se os 4 símbolos forem iguais a um deles à sua escolha.

10x Se os 4 símbolos forem iguais a outro símbolo.

20x Se os 4 símbolos forem iguais a outro deles.

50x Se os 4 símbolos forem iguais ao campeão escolhido por você.

? x escolha – se quiser – outras premiações. Só não se esqueça que o cassino (você!) só sobrevive se ganhar mais do que perder. Há um interessante capítulo da Estatística, chamada Teoria do valor esperado, que dá as ferramentas para esta análise. Um bom lugar para estudá-la é o belo livro "O andar do bêbado" de Leonard Mlodinov.

## HTML

Pode escrever o JS dentro do HTML, para diminuir um pouco a quantidade de arquivos (idem para o CSS, se quiser). Na tela, ter-se-á:



Prepare espaço para os seguintes elementos:

**creditos** O saldo do valor do apostador. Quando o sistema começa este campo é estabelecido com R\$100,00.

**aposta** A quantidade a ser apostada na próxima rodada. Inicia com R\$1,00

**simbolos** As 4 ocorrências obtidas após "rolar" os rolos.

**aposta -** O botão que diminui as apostas. Não pode ficar menos do que R\$1,00

**aposta +** Aumenta as apostas de um em um real

**jogar** simula o giro da alavanca.

**clique aqui** A chamada à tabela de premiações.

### Simulação do giro

Lembra das imagens GIF animadas? Aqui vamos criar uma dessas. Ache um editor de gifs (um bom está em [ezgif.com](http://ezgif.com)). Crie uma imagem chamada roleta.gif que apresente os 4 símbolos com intervalos de 20 milisseg entre cada um. Salve a imagem. Considere que os rolos param sempre na mesma ordem. O primeiro a parar é o mais à esquerda e o último é o mais à direita. A função JOGAR associada ao botão de mesmo nome:

1. Verifica se há créditos para realizar a aposta. Se não houver sinaliza o fato
2. Havendo, subtrai a aposta dos créditos e atualiza os 4 símbolos da tela com roleta.gif
3. Faz o seguinte 4 vezes:
  - Gera um aleatório entre 1..4
  - Dependendo do que der, atualiza a imagem correta do símbolo, mas antes manda esperar 4000 milisseg (4 segundos).
4. Para o segundo símbolo, espera 6000 milisseg.
5. Para o terceiro, espera 7000 mseg.
6. Para o quarto, espera 8000 mseg.
7. Verifica se houve ganho, depois de esperar 9000 mseg.

### Verificação da premiação

Crie uma matriz em JS

```
apos=[[1,1,1,1,5],  
      [2,2,2,10],[1,2,3,4,2],...];
```

cada linha desta matriz tem os 4 símbolos e o valor do prêmio se der esta configuração. A função que examina esta matriz é

```
function vepre(a, b, c, d){  
  para i de 0 ate apos.length  
    se a==apos[i][0] E b==apos[i][1] E  
      c==apos[i][2] E d==apos[i][3]  
      retorne apos[i][4]  
  fim{se}  
  fim{para}  
  retorne 0  
  fim{função}
```

## Dicas

1. Para limitar o tamanho do campo, faça

```
<input id = "credito" type = "text"  
size = "5" value = "R$ 0" readonly />
```

2. Para simular a pressão/liberação do botão, faça

```
<img src = "ima/apostamos.gif"  
onMouseDown="this.src='ima/ap. .nos_p.gif';"  
onMouseUp="this.src='ima/apostamos.gif';"  
onclick = "apostaMenos();" />
```

No HTML, a sintaxe `onMouseDown="this.src=..."` é utilizada para atribuir um novo valor à propriedade `src` de um elemento HTML quando o botão do mouse é pressionado sobre ele. O objeto `this`: Dentro do código JavaScript, a palavra-chave `this` se refere ao elemento HTML no qual o evento `mousedown` foi acionado.

3. Para mandar esperar faça

```
cont=setTimeout("document.  
getElementById('escud4').src  
= 'ima/operario.png'", 8000);
```

### 🔗 Para você fazer

Procure o arquivo ESQUELETO, que contém o CSS completo e partes do HTML e do JS necessários. Está tudo no mesmo arquivo, para simplificar a tarefa (e para você verificar que é possível fazer assim). Você precisará estudar este arquivo e promover as alterações/complementações que se fizerem necessárias.

Escreva o aplicativo aqui descrito. Mostre para o professor. Atente que haverá solicitações *ad hoc* durante a avaliação. (Por exemplo, crie uma premiação negativa, ou garanta que os símbolos serão iguais a uma configuração escolhida. Ou aumentar/diminuir a premiação, ou começar com 500 reais, ou ... . O céu é o limite. Avaliação:



307-76182 - gar a