

Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método sort() é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar sort().

Por padrão, sort() ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação: Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método substring(), usando expressões regulares, usando join(), mas aqui vai-se sugerir o uso de slice()). Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de slice é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são: **begin**: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end**: O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar. Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:

Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

71 13 98 50 1

e com o valor do troco igual a

69

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:
begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

10 5 40 6 1

e com o valor do troco igual a

70

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:
begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

98 3 45 20 1

e com o valor do troco igual a

52

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método sort() é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar sort().

Por padrão, sort() ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método substring(), usando expressões regulares, usando join(), mas aqui vai-se sugerir o uso de slice()). Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de slice é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:

begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:

Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Calcular

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

66 40 3 2 1

e com o valor do troco igual a

71

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação: Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são: **begin**: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end**: O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar. Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

9 15 20 10 1

e com o valor do troco igual a

84

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim(enquanto)
      i ← i + 1
    fim(enquanto)
  fim(enquanto)
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:
begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

45 9 21 18 1

e com o valor do troco igual a

90

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:

begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:

Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Calcular

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

21 35 40 13 1

e com o valor do troco igual a

74

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método sort() é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar sort().

Por padrão, sort() ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação: Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método substring(), usando expressões regulares, usando join(), mas aqui vai-se sugerir o uso de slice()). Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de slice é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são: **begin**: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end**: O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar. Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

71 21 6 40 1

e com o valor do troco igual a

62

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método sort() é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar sort().

Por padrão, sort() ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação: Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método substring(), usando expressões regulares, usando join(), mas aqui vai-se sugerir o uso de slice()). Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de slice é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são: **begin**: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end**: O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar. Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:

Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

18 35 50 5 1

e com o valor do troco igual a

65

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação: Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são: **begin**: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end**: O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar. Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:

Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

66 5 35 11 1

e com o valor do troco igual a

94

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método sort() é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar sort().

Por padrão, sort() ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método substring(), usando expressões regulares, usando join(), mas aqui vai-se sugerir o uso de slice()). Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de slice é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:

begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

10 2 15 9 1

e com o valor do troco igual a

55

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:

begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:

Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Calcular

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

21 71 6 3 1

e com o valor do troco igual a

98

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:

begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:

Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Calcular

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

20 21 40 13 1

e com o valor do troco igual a

68

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método sort() é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar sort().

Por padrão, sort() ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método substring(), usando expressões regulares, usando join(), mas aqui vai-se sugerir o uso de slice()). Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de slice é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:
begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

2 5 45 13 1

e com o valor do troco igual a

51

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método sort() é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar sort().

Por padrão, sort() ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação: Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método substring(), usando expressões regulares, usando join(), mas aqui vai-se sugerir o uso de slice()). Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de slice é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são: **begin**: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end**: O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar. Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

5 10 9 6 1

e com o valor do troco igual a

75

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:

begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:

Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

7 66 21 6 1

e com o valor do troco igual a

85

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:

begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

9 18 25 71 1

e com o valor do troco igual a

62

Responda aqui:

Moedas a devolver



Máquina de troco

Mais um exercício para desenvolver as habilidades até agora trabalhadas, a saber: HTML, CSS e finalmente a programação Javascript. O aplicativo é para calcular o troco em moedas.

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar um eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são 1, 6, 10, o troco de 19 centavos pode ser 10,6,1,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

O algoritmo

```
função troco (moedas[n], valor : inteiro)
  ordenar as moedas em ordem decrescente
  enquanto valor > 0
    i = 1
    enquanto i <= tamanho(moeda)
      enquanto valor >= moeda[i]
        escreva (moeda[i])
        valor = valor - moeda[i]
      fim{enquanto}
      i ← i + 1
    fim{enquanto}
  fim{enquanto}
fimfuncao
```

Ordenando Arrays com sort() O método `sort()` é o principal para ordenar arrays em JavaScript. Ele modifica o array original, então, se você precisar manter o array original, faça uma cópia antes de usar `sort()`.

Por padrão, `sort()` ordena elementos como strings. Para ordenar números corretamente, você precisa fornecer uma função de comparação:

Acompanhe o exemplo:

```
const numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b); // Ordem crescente
console.log(numbers); // Output: [1, 2, 3, 4, 5]

numbers.sort((a, b) => b - a); // Ordem decrescente
console.log(numbers); // Output: [5, 4, 3, 2, 1]
```

Dica para tirar a última vírgula Existem inúmeras maneiras de fazer isso (método `substring()`, usando expressões regulares, usando `join()`, mas aqui vai-se sugerir o uso de `slice()`. Acompanhe

```
ml= ml.slice(0, -1)
```

O uso de `slice` é o mesmo para arrays ou strings e ele foi copiado do ambiente Python. Ou seja, se você sabe usar lá, sabe aqui.

```
array.slice(begin, end)
string.slice(begin, end)
```

Seus parâmetros são:

begin: O índice onde a extração deve começar. Se for negativo, ele será contado a partir do final da matriz ou string. **end:** O índice um a menos do que o índice onde a extração deve terminar. Se for omitido, todos os elementos até o final serão extraídos. Se for negativo, ele será contado a partir do final da matriz ou string.

Retorna uma nova matriz ou string contendo a parte extraída. A matriz ou string original não é modificada.

```
const frutas = ['maçã', 'banana', 'laranja', 'uva'];
// Extrair os dois primeiros elementos
const primeirasFrutas = frutas.slice(0, 2);
// Extrair do segundo elemento até o final
const ultimasFrutas = frutas.slice(1);
// Extrair os dois últimos elementos
const duasUltimasFrutas = frutas.slice(-2);
```

```
Exemplo com strings:
const frase = "Olá, mundo!";
// Extrair a palavra "mundo"
const palavra = frase.slice(5);
// Extrair as primeiras 5 letras
const primeirasLetras = frase.slice(0, 5);
```

A encomenda

O aplicativo pede a sequência de moedas, separadas por vírgulas. Depois, pede-se o valor a devolver. O botão de calcular chama o programa que devolve a lista de moedas a dispensar.

Analise a tela do seu aplicativo

Apoio à folha h91

Geração de troco automático em moedas

- Forneça a série de moedas existentes no país
- Depois, forneça o valor do troco a ser produzido
- O seu programa indicará a sequência de moedas a serem fornecidas

Vamos lá

Preencha a data a seguir:
Moedas: (Moedas, separadas por vírgulas)

Valor a pagar: (nnn)

Moedas a liberar: (Moedas, separadas por vírgulas)

Para testar

1, 5, 10, 25, 50, 100 troco para 98 é: 50,25,10,10,1,1,1
1, 5, 10, 25, 50, 100 troco para 51 é: 50,1
1, 5, 10, 25, 50, 100 troco para 77 é: 50,25,1,1
1, 2, 5, 35, 50 troco para 59 é: 50,5,2,2
1, 7, 11, 20, 66 troco para 78 é: 66,11,1

o JS

```
function execu(){
  sm=document.getElementById('sm').value;
  val=parseInt(document.getElementById('val').value);
  sm=sm.split(',')
  sm=sm.map(Number)
  sm.sort((a,b) => b-a);
  ml= ' '
  while (val>0) {
    i=0
    while (i<=sm.length){
      while (val>=sm[i]){
        ml=ml+sm[i]+' '
        val=val-sm[i]
      }

      i=i+1
    }
  }
  ml= ml.slice(0, -1)
  document.getElementById('ml').value=ml
}
```

Avaliação:

Depois que programar o aplicativo calcule a sequência de moedas no país cujas moedas são

9 98 3 33 1

e com o valor do troco igual a

74

Responda aqui:

Moedas a devolver

