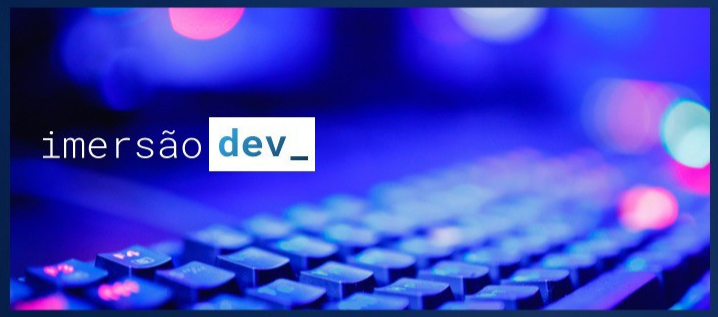
alura



Conhecendo HTML, CSS e Javascript_

Construa suas primeiras aplicações em 10 dias.

SOBRE ESTE LIVRO

A ideia de criar este livro veio durante a produção da Imersão Dev da Alura que aconteceu entre 22 de março e 2 de abril de 2021. A Imersão foi pensada para pessoas sem nenhum contato anterior com programação, e sabemos que esta primeira experiência pode ser decisiva! Assim, ficamos felizes em acompanhar quem está começando agora da melhor forma possível, e sabemos que isso pode ser feito de várias maneiras: síncrona, assíncrona, por vídeo e também por texto.

O conteúdo a seguir está totalmente baseado nos projetos desenvolvidos durante a Imersão Dev, mas conta com algumas explicações extras e diferentes. Ele foi pensado para servir de apoio ao conteúdo da Imersão, mas não é apenas uma transcrição dos vídeos e sim material extra em um formato mais adequado para o texto. Há várias formas de estudar e vários tipos de suporte para isso: vídeo, texto, áudio e inclusive papel. Parte do desafio do estudo de programação é justamente encontrar o que funciona melhor para você. Esperamos que este livro possa lhe ajudar de alguma forma a entrar com o pé certo nesse novo mundo.

Bons estudos!

SOBRE QUEM ESCREVEU

A produção dos capítulos foi dividida entre nosso time de conteúdo de programação:

Guilherme Lima Um dos condutores da Imersão Dev junto com a Rafaella, é desenvolvedor Python e instrutor back end na Alura.

Rafaella Ballerini Condutora da Imersão Dev com o Guilherme Lima, é instrutora front end da Alura.

Juliana Amoasei Desenvolvedora JavaScript e instrutora de programação back end na Alura.

AGRADECIMENTOS

A todas as 93.900 pessoas que se inscreveram para participar da Imersão Dev, motivadas pela mudança de carreira, reforço nos estudos ou, também, por que não, pela curiosidade de aprender algo novo. A Imersão foi feita para vocês, assim como todo este livro e os materiais extras.

A todos os times da Alura! Quem aparece são as devs e os devs, mas todas as imersões são produzidas por muitas mãos: time de devs que pensa e produz o conteúdo, didática que garante que o conteúdo faça sentido para quem está vendo de fora, edição e produção de vídeo, marketing, que faz com que tanta gente entre em contato com o evento e possa aprender programação com a gente, e o Scuba Team que dá suporte nas plataformas. Nossos agradecimentos!

COMO UTILIZAR ESTE LIVRO

Tudo que se refere a **trechos de código** neste livro utiliza uma das duas notações abaixo:

- trechos pequenos ou termos, como nomes de funções,

palavras-chaves ou expressões aparecerão no texto `desta forma`. Por exemplo, quando mencionamos uma função chamada `executarCodigo()` ou uma palavra-chave como `.length`.

- trechos maiores de código, como blocos de função, objetos ou similares aparecerão no texto em um parágrafo separado, como por exemplo:

```
function exibeTexto() {
   console.log("Olá Mundo!")
}
```

Todo o código será disponibilizado via link para consulta ou download (mais sobre isso abaixo), **mas é importante que você leia para entender o que acontece em cada parte!**

Sobre o GitHub

Utilizamos GitHub para disponibilizar o código finalizado de cada um dos projetos. O GitHub é uma plataforma de hospedagem de códigos que permite que qualquer pessoa cadastrada possa acessar e contribuir com projetos de código aberto.

O Github representa todo um universo ligado à programação, o chamado **controle de versão ou versionamento de código**. É uma das ferramentas principais no dia-a-dia de quem trabalha na área e você pode ir aprendendo como utilizá-la aos poucos.

Ao final de cada capítulo, você terá acesso ao código completo disponibilizado no GitHub, com a seguinte estrutura:

```
app.js
index.html
style.css
```

Sinta-se a vontade para executar os programas no Codepen ou em algum editor de código de sua preferência.

Redes sociais

Paulo Silveira

- Instagram
- Twitter
- Linkedin

Rafaella Ballerini

- Instagram
- Linkedin
- Youtube

Guilherme Lima

- Instagram
- Youtube
- Linkedin

Juliana Amoasei

- Linkedin

Sumário

Versão: 25.5.6

CAPÍTULO 1

CONVERSOR DE MOEDAS

1.1 OBJETIVO DA AULA

Nesta primeira aula, o objetivo é criar um programa que peça para o usuário fornecer um valor em dólar, e a partir desse valor o programa vai exibir na tela o equivalente em real.

Neste programa focamos nas primeiras ferramentas principais de qualquer linguagem de programação: variáveis e operadores, além das funções `alert` e `prompt` para trocar nossas primeiras mensagens na tela com o usuário.

1.2 PASSO A PASSO

Antes de começarmos a escrever qualquer linha de código, temos que pensar no caminho que o programa deve percorrer e quais informações ele precisa para ser executado corretamente. No caso do nosso conversor de moedas:

1. O que o programa deve fazer? Converter dólar para real…

Ok, mas como isso é feito? Se fôssemos converter 10 dólares para real "na mão", faríamos a seguinte conta: `10 * 5.50 =`

`55.00`.

Onde `10` é o valor em reais que queremos converter e `5.50` é a taxa média do dólar no momento em que escrevemos este texto. Multiplicando um valor pelo outro, temos o resultado em reais: `55.00`.

2. Como o programa vai receber essas informações?

Podemos concluir, a partir do que vimos no ponto 1, que os dados que o programa precisa para funcionar são:

- o valor em dólar que queremos converter (será fornecido pelo usuário);
- a taxa do dólar que será utilizada na conversão (que nós vamos fornecer ao programa no momento em que escrevemos o código).

3. Qual o "resultado" do programa:

Após executar todo o código, o programa deve exibir ao usuário um número que representa o resultado da operação `valor em dólar * taxa de câmbio`.

Agora que já sabemos quais dados o programa precisa para funcionar, o que ele precisa fazer e qual o resultado final dele, podemos começar a escrever código!

1.3 ESCREVENDO O PROGRAMA

Já que precisamos receber do usuário o valor em dólar que ele quer converter, a primeira coisa a fazer é "pegar" esse valor de

alguma forma. Vamos utilizar uma ferramenta do JavaScript chamada `prompt`, que abre no navegador uma caixa de texto onde o usuário pode inserir informações. Utilizamos os parênteses `( )` para passar para o usuário uma mensagem, assim ele sabe qual informação estamos pedindo **— não esqueça das aspas!**.

```
var valorEmDolarTexto = prompt("Qual o valor em dolar que você qu
er converter?")
```

É muito importante sempre lembrar de "guardar" todos os dados que queremos usar em nossos programas e, para isso, usamos variáveis. Variáveis são como pequenos espaços de memória onde guardamos dados, como por exemplo um número ou um texto. Nesse caso, criamos com a palavra-chave `var` uma variável chamada `valorEmDolarTexto` que vai "guardar" a informação que o usuário vai passar para o programa. E que informação é essa? O valor em dólar para ser calculado.

> O nome da variável é muito importante! É através dele que o programa consegue identificar e localizar todos os dados que "guardamos" para serem utilizados pelo programa, e que podem ser muitos!

Agora que já temos o valor em dólar, vamos esbarrar em um probleminha muito comum e característico da programação, que é a distinção entre textos e números. Por padrão, quando usamos o `prompt` para pedir alguma informação ao usuário, o JavaScript **sempre** interpreta essa informação como texto, não importa se o usuário escreve `2`. Essa é uma questão relacionada a *tipos de dados em computação*, que neste momento não vamos detalhar

muito, mas você pode pesquisar para saber mais.

Já que é assim, se o usuário inserir o valor `10` para converter, o JavaScript pode ficar confuso, afinal de contas ele não sabe que `"10"` é um número, e como é que ele vai fazer essa multiplicação?

Felizmente, como já dissemos, essa questão *texto x número* é comum. E o próprio JavaScript tem uma palavra-chave pronta para fazer essa conversão. Então, antes de qualquer outra coisa, devemos pegar o valor que está guardado na variável `valorEmDolarTexto` e transformá-lo de texto para número:

```
var valorEmDolarTexto = prompt("Qual o valor em dolar que você qu
er converter?")

var valorEmDolarNumero = parseFloat(valorEmDolarTexto)
```

A palavra-chave `parseFloat()` é essa ferramenta pronta do JavaScript para converter o que está entre os parênteses de texto para número. No caso, o que vai ser convertido é o valor que o usuário vai inserir no programa através do `prompt` e que está guardado na variável `valorEmDolarTexto`.

Como sempre, precisamos guardar em uma variável esse valor convertido, para que o programa possa usá-lo. Já que agora o valor é um número, chamamos essa nova variável de `valorEmDolarNumero`.

Já podemos fazer a conta? E a taxa do dólar? Há algumas formas de passarmos para o programa a informação da taxa do dólar. Para esta primeira versão do nosso conversor vamos dizer ao JavaScript exatamente qual é a taxa e utilizá-la na conta de multiplicação:

```
var valorEmDolarTexto = prompt("Qual o valor em dolar que você qu
```



```
er converter?")

var valorEmDolarNumero = parseFloat(valorEmDolarTexto)

var valorEmReal = valorEmDolarNumero * 5.50
```

Sem esquecer de "guardar" o valor da multiplicação em uma variável, afinal de contas precisamos dele para exibir ao usuário! Neste caso, a taxa utilizada é `5.50`; se lembrarmos que a variável `valorEmDolarNumero` já está guardando um número (informado pelo usuário), o JavaScript consegue fazer a substituição e usar os valores para fazer a conta :)

Estamos guardando o resultado desta conta na variável `valorEmReal`.

Uma última coisa antes de exibir o resultado para o usuário! Quando fazemos contas que envolvem valores "quebrados" — ou seja, com um ou mais dígitos depois da vírgula — pode acontecer a tal da *dízima periódica*, ou o número após a vírgula ficar com muitas casas decimais. Então vamos usar uma outra palavra-chave que o JavaScript já deixou pronta para esses casos:

```
var valorEmDolarTexto = prompt("Qual o valor em dolar que você qu
er converter?")

var valorEmDolarNumero = parseFloat(valorEmDolarTexto)

var valorEmReal = valorEmDolarNumero * 5.50
var valorEmRealFixado = valorEmReal.toFixed(2)
```

Essa palavra-chave é `.toFixed()` e deve ser usada da forma que está no exemplo: nome da variável, ponto, `toFixed` e entre parênteses vai um número com as casas decimais que queremos arredondar.

Ou seja, `nomeDaVariavel.toFixed(2)` vai arredondar o

número que está guardado dentro da variável `nomeDaVariavel`. Se for o número `5.77777777` ele vai ser arredondado para `5.77`. Sem esquecer de guardar esse novo valor em uma variável! No caso, usamos a variável `valorEmRealFixado` para isso.

Agora já está tudo pronto para exibir o resultado para o usuário. Podemos usar a ferramenta `alert()` que vai exibir a informação em uma caixa de texto no navegador. Só precisamos passar essa informação dentro dos parênteses; se informarmos o nome da variável onde este dado está guardado, o JavaScript consegue fazer a substituição:

```
var valorEmDolarTexto = prompt("Qual o valor em dolar que você qu
er converter?")

var valorEmDolarNumero = parseFloat(valorEmDolarTexto)

var valorEmReal = valorEmDolarNumero * 5.50
var valorEmRealFixado = valorEmReal.toFixed(2)

alert(valorEmRealFixado)
```

Alguns detalhes que sempre temos que lembrar:

- Podemos dar o nome que quisermos para as variáveis, porém o JavaScript tem algumas regras: não comece com letra maiúscula nem com números, não use espaços, caracteres especiais e nem acentos. Escolha um nome que descreva o que a variável vai "guardar"! O padrão de nomes do JavaScript é `nomeDaVariavel`, usando maiúsculas para diferenciar as palavras (mas nunca no início da palavra).
- O JavaScript tem muitos trechos de código prontos para utilizarmos em nossos programas, como por exemplo `parseFloat()` para converter um texto em número e `variavel.toFixed(2)` para arredondar. Estes códigos estão sempre escritos em inglês e devem ser usados exatamente como estão nos exemplos para funcionar — não coloque pontos nem espaços onde não tem, não troque maiúsculas por minúsculas e não esqueça dos parênteses nos lugares certos, pois esses pequenos detalhes fazem com que o JavaScript "entenda" o que está sendo escrito. Com prática e estudo você vai entender melhor para que servem e o que significam os pontos, parênteses e tudo mais.

Tudo certo? Então é hora de praticar!

Clique neste link para acessar o código completo no GitHub.

1.4 RESUMO

Vamos repassar todos os passos do programa:

- Obter do usuário o valor que vai ser convertido
- Converter esse dado para número
- Fazer a operação matemática (multiplicação)
- Arredondar o resultado
- Exibir o resultado na tela para o usuário

Capítulo 2

CALCULADORA

2.1 OBJETIVO DA AULA

Criar um programa que pergunte e armazene dois valores e o usuário escolha qual operação deseja realizar com um número. Por exemplo: para somar a pessoa teria que digitar 1, para subtrair, 2, etc. **Neste programa, focamos em laços condicionais, operações booleanas, if e else**.

2.2 PASSO A PASSO

Vamos começar criando uma variável para armazenar o primeiro valor.

```
var primeiroValor
```

Agora vamos pedir para o usuário digitar o primeiro valor.

```
var primeiroValor = prompt("Digite o primeiro valor:")
```

Como queremos realizar operações matemáticas com esses valores, vamos converter o primeiro valor em um número inteiro com o `parseInt` .

```
var primeiroValor =prompt("Digite o primeiro valor:")
primeiroValor = parseInt(primeiroValor)
```


> Dica: essa conversão pode ser feita em apenas uma linha. Podemos usar as funções `prompt` e `parseInt` na mesma linha, e teremos o mesmo resultado.

```
var primeiroValor = parseInt(prompt("Digite o primeiro valor:"))
```

Agora, vamos criar uma segunda variável para armazenar o segundo valor.

```
var primeiroValor = parseInt(prompt("Digite o primeiro valor:"))
var segundoValor = parseInt(prompt("Digite o segundo valor:"))
```

Agora que temos dois valores, precisamos perguntar qual operação será realizada (se multiplicação, soma, divisão ou subtração). Para isso, vamos representar cada operação com um número, como ilustra a tabela abaixo.

número da operação	operação
1	Divisão
2	Multiplicação
3	Soma
4	Subtração

Sabendo disso, vamos criar uma variável para armazenar o número da operação escolhida através da função `prompt` .

```
var operacao = prompt("Digite 1 para fazer uma divisão, 2 para mu
ltiplicação, 3 para soma e 4 para subtração: ")
```

Agora precisamos saber **se** a variável `operacao` é igual a 1. No JavaScript e em muitas outras linguagens, essa condição **se** é escrita

no idioma inglês **if** (que significa “se” em português).

```
if
```

Para criar a condição que verifica **se** a variável `operacao` é igual a 1 vamos incluir o sinal de parênteses.

```
if ()
```

Dentro dos parênteses, vamos incluir nossa condição, e aqui tem um detalhe importante. Comparar **se** a variável `operacao` é **igual** a **1** não será feita com apenas um sinal de igual, e sim com dois (`==`)..

```
if (operacao == 1)
```

Isso acontece porque quando usamos apenas um sinal de igual, estamos atribuindo um valor a uma variável, e não é o que queremos fazer. Queremos comparar se o valor da variável operação é igual a um número, neste caso, 1.

> Dica: com um sinal de igual (`=`), estamos atribuindo um valor a uma variável (`var nome = "Maria"` , por exemplo). Com dois sinais de igual (`==`), estamos comparando dois valores.

Caso isso seja verdadeiro, isto é, caso a operação escolhida seja *igual igual* a 1, queremos fazer alguma coisa. Esse fazer alguma coisa será indicado pelos sinais de abrir e fechar chaves (`{ }`).

```
if (operacao == 1) {

}
```


Ao fazermos isso, todo o código escrito dentro dos “bigodes”, quer dizer, chaves, será executado caso a operação seja igual a 1. Dentro das chaves, precisamos dividir o valor armazenado na variável `primeiroValor` pelo `segundoValor` e guardar esse resultado em algum lugar. Podemos criar uma nova variável chamada `resultado` .

```
if (operacao == 1) {
  var resultado
}
```

Na programação, as operações são representadas de formas diferentes. Veja isso no quadro abaixo.

símbolo na programação	operação
/	Divisão
*	Multiplicação
+	Soma
-	Subtração

Sabendo disso, podemos usar a barra para dividir o primeiro valor pelo segundo valor.

```
if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
}
```

Vamos executar nosso programa, atribuir os valores `10` no primeiro valor e `2` no segundo, selecionar a opção 1, e ver o que acontece?

Nada :(Não aconteceu nada. O resultado da operação não apareceu na tela e isso é triste. Podemos incluir um alert exibindo o resultado.

```
if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
  alert(resultado)
}
```

Vamos testar agora?

Deu certo :)

Maaaaas podemos fazer algo melhor: no lugar de exibirmos um *alert*, que tal escrevermos na tela? Que tal exibirmos o resultado da divisão na tela?

Para isso, podemos usar um outro comando, chamado `document.write`, e passar o valor que queremos exibir entre os parênteses.

```
if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
  document.write(resultado)
}
```

Agora sim! O resultado apareceu na tela e ficou bem legal. Podemos melhorar ainda mais.

Que tal incluirmos os valores das variáveis `primeiroValor` e `segundoValor` com o símbolo de divisão entre elas, e o sinal de igual informando o resultado? Para juntar os valores das variáveis com os símbolos do tipo texto, usaremos o sinal de soma (`+`), e as aspas para os símbolos, assim:

```
if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
  document.write(primeiroValor + " / " + segundoValor + " = " + r
esultado)
}
```


> Dica: dentro do `document.write`, podemos incluir as tags do HTML e estilizá-las no CSS, alterando o tamanho, posicionamento, cores e muitas outras coisas.

A equipe da Alura estilizou uma tag para exibir a operação bem bonita na tela. A tag usada foi o `<h2>`. Podemos incluí-lo no `document.write` e realizar um novo teste.

```
if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
  document.write("<h2>" + primeiroValor + " / " + segundoValor +
" = " + resultado + "</h2>")
}
```

Ficou sensacional, mas temos um problema. Nossa calculadora realiza apenas a divisão quando a operação é 1. Mas e as outras operações?

Podemos verificar o caso de **se** a operação **não** for igual a `1`. Esse **se não** é chamado por meio do comando `else`.

```
if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
  document.write("<h2>" + primeiroValor + " / " + segundoValor +
" = " + resultado + "</h2>")
} else
```

Agora, precisamos de uma nova condição: comparar **se** a variável `operacao` é **igual** a `2`, realizar a operação de multiplicação com o símbolo do asterisco (`*`) e alterar o texto do `document.write`:

```
if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
```

```
  document.write("<h2>" + primeiroValor + " / " + segundoValor +
" = " + resultado + "</h2>")
} else if (operacao == 2) {
  var resultado = primeiroValor * segundoValor
  document.write("<h2>" + primeiroValor + " x " + segundoValor +
" = " + resultado + "</h2>")
}
```

Podemos realizar o mesmo para as operações soma e subtração:

```
if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
  document.write("<h2>" + primeiroValor + " / " + segundoValor +
" = " + resultado + "</h2>")
} else if (operacao == 2) {
  var resultado = primeiroValor * segundoValor
  document.write("<h2>" + primeiroValor + " x " + segundoValor +
" = " + resultado + "</h2>")
} else if (operacao == 3) {
  var resultado = primeiroValor + segundoValor
  document.write("<h2>" + primeiroValor + " + " + segundoValor +
" = " + resultado + "</h2>")
} else if (operacao == 4) {
  var resultado = primeiroValor - segundoValor
  document.write("<h2>" + primeiroValor + " - " + segundoValor +
" = " + resultado + "</h2>")
}
```

Para finalizar, pense no seguinte cenário: assim que o programa inicia, uma pessoa entra com `10` no primeiro valor, `2` no segundo valor e escolhe a opção **5**. O que será que vai acontecer?

Nosso calculadora realiza quatro operações, e não temos a opção **5**. Quando esta opção for escolhida, nada vai acontecer. Podemos incluir uma condição informando que a operação é inválida caso a opção dada seja diferente de 1, 2, 3 e 4.

Para isso, vamos usar apenas o comando `else` . Ou seja, se a operação não for com 1, 2, 3 e nem com 4, nossa calculadora indicará "opção inválida", sem precisar saber qual opção foi atribuída.

```
else {
  document.write("<h2>Opção inválida</h2>")
}
```

Aqui está o código completo desta aula:

```
var primeiroValor = parseInt(prompt("Digite o primeiro valor:"))
var segundoValor = parseInt(prompt("Digite o segundo valor:"))

var operacao = prompt("Digite 1 para fazer uma divisão, 2 para mu
ltiplicação, 3 para soma e 4 para subtração: ")

if (operacao == 1) {
  var resultado = primeiroValor / segundoValor
  document.write("<h2>" + primeiroValor + " / " + segundoValor +
" = " + resultado + "</h2>")
} else if (operacao == 2) {
  var resultado = primeiroValor * segundoValor
  document.write("<h2>" + primeiroValor + " x " + segundoValor +
" = " + resultado + "</h2>")
} else if (operacao == 3) {
  var resultado = primeiroValor + segundoValor
  document.write("<h2>" + primeiroValor + " + " + segundoValor +
" = " + resultado + "</h2>")
} else if (operacao == 4) {
  var resultado = primeiroValor - segundoValor
  document.write("<h2>" + primeiroValor + " - " + segundoValor +
" = " + resultado + "</h2>")
} else {
  document.write("<h2>Opção inválida</h2>")
}

// escrevendo na tela - document.write()
// concatenação (juntar palavra com variáveis) - ("palavra" + var
iavel)
// == comparação é diferente do = que usamos para fazer atribuiçã
o

// if = se
```

```
// else = se não
// else if = se não se
```

Tudo certo? Então é hora de praticar!

Clique neste link para acessar o código completo no GitHub.

2.3 RESUMO

Vamos repassar todos os passos do programa:

1. Interagir com usuário para receber os valores que serão calculados;
2. Criar a lógica para saber qual operação será executada;
3. Comparar o valor da operação escolhida para descobrir qual função será realizada;
4. Exibir uma mensagem de erro, caso a operação escolhida seja inválida;
5. Exibindo os valores escolhidos e o resultado da operação com document.write;

CAPÍTULO 3

MENTALISTA

3.1 OBJETIVO DA AULA

Criar um programa que sorteia um número secreto entre 0 e 10, em que o usuário possui três tentativas para acertar esse número. Para auxiliar quem joga, quando o chute é incorreto dizemos se ele é maior ou menor que o número secreto. Quando o chute é correto, informamos o fim do jogo e o valor do número secreto. **Neste programa, focamos no uso do while, condicionais e comparação entre variáveis**.

3.2 PASSO A PASSO

Para iniciar esse programa, vamos criar uma variável chamada numeroSecreto e atribuir um número aleatório entre 0 e 10. Vamos utilizar uma função chamada `Math.random` :

```
> Math.random()
<- 0.20982489919137293

> Math.random()
<- 0.43822764751936005

> Math.random()
<- 0.2983069465358348

> Math.random()
```