

Imersão Dev - Aula 1

Rafaella: Olá! Eu sou a Rafaella Ballerini e sou desenvolvedora e instrutora front-end aqui na Alura. Eu, Gui Lima e Paulo Silveira vamos te acompanhar nessas duas semanas de Imersão Dev. Eu estou muito ansiosa por essa imersão, pois você não apenas aprenderá conteúdo de lógica de programação, mas também vai sair daqui com vários projetos para colocar em seu portfólio.

O legal desses projetos é que todos podem ser personalizados da forma que você quiser, alterando as cores, a fonte, o texto e assim por diante, permitindo que você abuse da sua criatividade.

Para isso, vamos utilizar uma ferramenta da internet, que é o CodePen. Sendo assim, você não precisa se preocupar em baixar e instalar linguagens ou editores de texto. Você pode se perguntar se vai aprender a programar de fato ou se só veremos teoria de lógica da programação, e sim, você vai aprender a programar! Não somente isso, eu já quero te chamar de programador ou programadora desde esse primeiro projeto.

Justamente por isso, é importante que você coloque todos os projetos que desenvolver nas redes sociais, no LinkedIn, no Instagram e até no GitHub. Caso você não conheça essa ferramenta, não tem problema, pois vamos colocar alguns conteúdos extras (inclusive do canal da Alura e do meu canal no Youtube) para você acompanhar tranquilamente.

Outra coisa legal dessa imersão é que temos uma comunidade no Discord própria para todo mundo que está participando. Se você quiser tirar alguma dúvida, conversar conosco ou mesmo mostrar os projetos que têm feito, não deixe de participar da comunidade. Pegue o seu café, abra o CodePen e vamos para a primeira aula.

Estou aqui com o Gui Lima, tudo bem com você?

Guilherme: Oi Rafa, tudo bem gente?

Rafaella: E com o Paulo Silveira, o CEO da Alura. E ele gostaria de conversar um pouco com vocês também!

Paulo: Oi Rafa, oi Guilherme. Eu estou muito contente de fazermos mais uma edição da Imersão Dev, esse evento que já ficou marcado para a comunidade. Tem muita gente que entrou na carreira de desenvolvimento através dessa imersão. Está sendo muito impressionante o que temos visto. É óbvio que esses 10 dias que você estará estudando conosco, praticando e desenvolvendo, serão intensos e o tempo é curto.

Não será trivial essa entrada na carreira de dev, mas você verá que é um pouco menos difícil do que você imaginava e que, se você realmente optar por essa carreira de desenvolvimento de software, com tempo e dedicação você sem dúvidas conseguirá

chegar lá. Antes de eu mostrar depoimentos de pessoas que estiveram no seu lugar, estudaram conosco na Imersão Dev e hoje estão trabalhando na área (depois de muito esforço e dedicação), eu vou jogar uma pergunta para a Rafa e o Guilherme.

Eu sou formado em Ciência da Computação. Quando a gente fundou a empresa, nós estávamos muito focados em cursos mais avançados. Dado que o mercado cresceu tanto, fomos abraçando também as pessoas que estão querendo entrar na comunidade e começar a sua carreira, pois sabemos que esses primeiros passos às vezes podem assustar.

A minha missão com a Rafaella e o Guilherme aqui é tirar esses obstáculos e barreiras para que você perceba que consegue fazer os conteúdos apresentados aqui, fazer os desafios e ir se aprimorando, conversando no Discord e mostrando no LinkedIn e no Instagram o que está fazendo.

Rafaella, conta para nós o que você estudava antes e como você veio parar na programação, em criar aplicações, sites e sistemas, que é o que você aluno ou aluna vai fazer com a gente.

Rafa: Eu era da medicina, não era tão fã do que eu fazia, achava que a Medicina tinha um pouco da "decoreba" que acredito que nós, pessoas que gostam de desenvolver softwares, não gostam muito. Nós gostamos de algo mais lógico, de entender o que está acontecendo, não só decorar. Acabei migrando de área, foi um processo um pouco longo, eu estava no terceiro ano da medicina, mas me encontrei quando comecei a entender o que eram algoritmos.

No terceiro ano eu nem sabia que existia essa possibilidade de programar, desenvolver, não era apresentado para mim. Quando eu descobri toda a lógica que existe por trás, como tudo funciona no computador, eu me encantei. E acredito que você também vai se encantar, pois é uma área maravilhosa.

Paulo: A Rafa é o primeiro exemplo dessas pessoas que estão navegando em outras carreiras, estudando outras coisas, e caem no desenvolvimento de software. E essa história é cada vez mais comum, por vários motivos. Primeiro porque a gente realmente gosta do código, gosta de criar. Depois vamos ver que também existe uma parte bem repetitiva na programação, não é tudo tão bonito, mas tem muito da criatividade de vocês.

Por isso, aqui na imersão, todas as dez aulas estão muito bem planejadas para que você consiga levar com a sua cara, o seu tom, as suas preferências. Você vai perceber que é possível criar coisas completamente diferentes do que o Gui e a Rafa apresentarão aqui. Você sairá da aula com um endereço na internet e poderá mostrar para toda sua família que foi você quem fez e que aquilo tem a sua cara.

Não é o caso da Rafa, mas tem muitos médicos e médicas que estão apresentando programação por conta da ciência de dados, pela possibilidade de usar aparelhagem, laudos, mecanismos e aplicações para automatizar algumas coisas. O software vai

"comendo o mundo", uma expressão famosa de alguns investidores que implica que a programação está tomando algumas carreiras. Não todo mundo, mas muita gente vai precisar saber um pouquinho de programação para melhorar ainda mais a própria profissão.

Guilherme, você quer contar a sua história de como veio parar no código?

Guilherme: Foi meio maluco, porque comecei a estudar piano aos 7 anos, cresci como músico e comecei a dar aula de música aos 14 anos. Porém, eu tive uma frustração, passei em uma prova específica de música, mas não consegui me inscrever, e isso me deixou muito triste. Nesse ponto eu decidi estudar outra coisa, ingressar em outra área no mercado.

Em uma conversa totalmente aleatória com uma amiga da minha mãe, que trabalhava na universidade onde me formei em Sistemas da Informação, ela disse que eu me daria bem em TI. Ela começou a falar sobre sistemas de informação, mexer com computador, criar programas e aplicativos. Eu achei interessante e me inscrevi.

Minha primeira aula foi algoritmo, e eu fiquei perdido. A professora perguntava como fazemos para realizar determinada tarefa, e a sala inteira quebrava a cabeça. Nesse primeiro dia, sem brincadeira, um colega que conheci durante o intervalo desistiu, foi embora. É um susto mesmo, quando a gente ouve a palavra "algoritmo" já pensa ser coisa de maluco.

Com muita dedicação, muita garra, muitas noites sem dormir, consegui me formar. Hoje estamos aqui para compartilhar e fazer desse primeiro momento uma experiência um pouco mais agradável. Essa primeira aula não vai ser de algoritmo, vai ser muito mais alegre e descontraída, e espero que vocês gostem e que a jornada de vocês seja inspiradora para outras pessoas também.

Paulo: E eu também comecei na carreira com um interesse que nós três temos em comum, que são os jogos de videogame. Tem muita gente que vem para a programação pois quer criar o próprio jogo. A maioria não consegue, mas acaba entrando nessa carreira. No desenvolvimento de jogos, desde os mais simples, até os mods para adaptar cenários ou scriptar eventos, é bem comum ter que escrever código em uma linguagem que o próprio jogo fornece, muitas delas parecidas com a linguagem que vamos usar, que é o Javascript.

Se você gosta de charadas, de lógica de programação, de board games, de puzzles ou brincadeiras matemáticas, essa imersão também é para você. E nós queremos que você estude junto com seus amigos ou amigas, então ainda dá tempo de você convidar aquela pessoa que sempre se interessou por programação para seguir os estudos com você. Meu pedido é que você indique a Imersão Dev para mais uma pessoa e que ela se inscreva hoje, reservando duas horas de dedicação por dia, pois o conteúdo não é trivial.

Ainda teremos uma galeria com os projetos de vocês, e vocês poderão votar nos projetos que mais gostaram. Para darmos uma última prova antes de começarmos a escrever código, vou pedir para a produção colocar aqui no vídeo alguns depoimentos incríveis de alunos e alunas que começaram a carreira por causa da Imersão Dev.

Claro que poderia ter sido de outra forma, mas queremos mostrar que esse pode ser um momento especial para sua carreira e sua vida. Nós queremos que você, daqui a 3 anos, esteja trabalhando em uma boa empresa e que esse tenha sido o pontapé inicial. Aqui, além de programadores, somos professores e professoras, e a nossa profissão é ensinar. Esse é um orgulho muito forte, tocar as pessoas. Mais do que dar uma aula incrível ou ser muito bom no código, o importante é sermos memoráveis para os alunos e alunas.

Vamos para os depoimentos, produção!

Depoimento 1: As pessoas se predispunham a ajudar, a gente não tinha que recorrer à equipe, dando dicas de como resolver problemas ou chegar em resultados melhores, ainda que por caminhos mais difíceis. Achei muito legal, e você se sente acolhido. Eu vinha de trabalhos onde eu tinha que resolver tudo sozinha, não que eu não conseguisse ou não gostasse, mas acredito que poder recorrer a outras pessoas e compartilhar conhecimento vai me agregar e me fazer crescer muito mais como profissional.

Também acredito que eu vá poder contribuir para outras pessoas, por mais que eu seja iniciante. Essa comunidade foi muito boa, muito legal. Você se sente capaz, eu me senti capaz de resolver os problemas, seja de um jeito ou de outro. Isso dá um pouco de fôlego e você consegue seguir em frente.

É uma experiência legal que nos deixa confiantes de que, com dedicação, chegaremos a algum lugar.

Depoimento 2: Tive algumas outras experiências tentando aprender desenvolvimento web, coisas básicas como HTML e CSS, não na Alura, mas não deu muito certo, eu não consegui reter o conhecimento e foi bem frustrante. Isso somado a todos os outros problemas, eu já estava em vias de desistir, de dar um passo atrás na carreira e pensar em outras opções.

Eu lembrei da Imersão Dev e decidi tentar mais uma vez, e ainda nem que eu participei! Eu adorei a didática deles, a forma como passaram o conhecimento. Eu sei que foram projetos simples, mas a forma como foi passado me permitiu aprender algo que eu já tinha tentado antes, mas que eu não tinha conseguido reter.

Me diverti fazendo aquilo. Teve dias que fiquei até 2, 3 horas da manhã estudando sem perceber. Eu fiz contatos na Imersão, inclusive com outros programadores. Foi algo que fiz como última tentativa e deu muito certo, estou muito animado para continuar estudando.

Paulo: Para você ver que a programação está em todo lugar, todos os sites, todas as páginas web que você acessa, foram criados com código. O Guilherme está compartilhando a tela para vermos um exemplo, que é o canal do [Youtube da Alura](#), e já fica o convite para você se inscrever.

Todos esses elementos na tela, como thumbnails, mouseovers (quando passamos o mouse em cima e é exibida uma mensagem), foram desenvolvidos com código. Alguém escreveu código para que esses elementos possam reagir de acordo com a interação com o aquilo que você está fazendo.

Nessa imersão você não vai precisar instalar nada. Vamos ter vídeos e tarefas opcionais para ir além, mas para acompanhar a imersão basta ter o Chrome instalado para aprender a programar e dar esse mergulho na sua carreira.

Se você der um clique com o botão direito no fundo da tela do Youtube e acessar o menu "Inspecionar" (cujo atalho é "Ctrl + F12"), ele vai abrir uma aba na lateral ou na parte inferior da tela mostrando como os elementos visuais foram codificados em uma "linguagem", por assim dizer, chamada HTML.

Nós vamos passar por ela, mas nesse comecinho, para programarmos mesmo, vamos nos concentrar em uma linguagem chamada Javascript, que também está presente no meio desse código. Inclusive, o Guilherme está no submenu "elementos", mas também temos o submenu "console", que será o seu melhor amigo durante essa imersão.

Você vai consultá-lo muitas vezes e ver várias mensagens de erro em amarelo, em vermelho, em inglês e assim por diante. Lembre que no começo do vídeo eu falei que nosso objetivo é remover obstáculos, então não tenha medo do inglês, das mensagens de erro ou de eventuais diferenças na sua tela.

É muito comum trabalharmos com incertezas na programação, e raramente as coisas que encontramos na internet são exatamente aquilo que precisamos para resolver um problema. E isso é muito legal, pois abre margem para criatividade na hora enfrentar os desafios.

Na parte inferior do console, se você reparar, tem uma setinha e um cursor piscando, o que normalmente significa que podemos colocar algum input de programação para ser executado. O console do Chrome não é onde normalmente programamos no dia-a-dia, e nessa imersão vamos usar um site especial para isso.

Porém, somente para testarmos, se escrevermos uma expressão lógica nesse console, o computador - na verdade o navegador - vai tentar resolver. Um exemplo seria `5 + 5`. Vamos digitar essa expressão e pressionar "Enter" para vermos o que acontece.

`5 + 5`

Paulo: Como resultado, teremos `10`. Então você pode rodar expressões, cálculos e várias outras coisas nesse console. Além disso, você pode rodar "comandos", que também podemos chamar de funções ou métodos. Rafa, que comando podemos escrever para não assustar muito os alunos e alunas?

Rafaella: Acho que o próprio `console.log()`,.

Paulo: Acho bom. Esse é o comando que usamos para imprimir alguma coisa no console, como uma frase, "Boas-vindas à imersão dev".

```
console.log("Boas-vindas à imersão dev")
```

Paulo: Se a gente der um "Enter", vai aparecer a frase embaixo, além de um "undefined" que a função retornou. Tem um monte de coisa acontecendo aqui que não tem problema você não saber, e inclusive esperamos que você não saiba. Se você está testando isso em casa para brincar, tem um monte de detalhes que você precisa tomar cuidado para não se assustar, como os parênteses, as aspas duplas e assim por diante, detalhes de sintaxe que precisam ser respeitados, além de outros que não.

Na programação você sempre vai encontrar esses cenários. Às vezes algo parecido com isso funciona, e às vezes não. Quando você está acessando um site pelo navegador, você pode sempre inspecionar os elementos para entender como aquela página foi construída - essa parte mais visual, que chamamos de "Front-end".

Inspecionando essa parte onde conseguimos digitar e clicar, que é a "frente" do site, conseguimos ver o código que as pessoas escreveram para desenhar, colocar os elementos na tela e interagir com você. Isso já é um sistema, e existem inúmeras vagas de trabalho somente para trabalhar nessa área, que está mais próxima do que a Rafa trabalha. Existem várias especialidades nas carreiras de programação e desenvolvimento.

Rafaella: Agora que já entendemos um pouco de como uma página se parece, vimos um pouco de código e o console, acho que podemos migrar para o site que será nosso parceiro aqui na Imersão Dev, o CodePen. Ele é uma plataforma onde você pode escrever código HTML, CSS e Javascript, e ele vai interpretar esses códigos e exibir na tela o resultado deles funcionando em conjunto, parecido com o que estávamos fazendo no console.

É interessante se cadastrar na plataforma para permitir a criação de um novo "Pen", que é o nome dado aos projetos feitos no site. Feito isso, já poderemos começar a programar.

Paulo: Na primeira vez que você acessar o CodePen, que é em inglês (não se assuste com isso!), você verá um botão para criar sua conta, que é gratuita para a maior parte das funcionalidades. É no CodePen que você vai escrever os seus códigos e compartilhar com amigos e, especialmente, comigo, com a Rafa e com o Guilherme, catalogando todos os passos ao longo do projeto.

Rafa e Guilherme, vocês podem mostrar como criar a conta?

Guilherme: Em <https://codepen.io/>, clicamos em "Sign Up For Free" para realizar um cadastro gratuito. Temos então quatro opções: vincular uma conta com o Twitter, com o GitHub, com o Facebook ou com um e-mail. Clicando em "Sign Up with Email", por exemplo, basta preencher os campos com nome, nome de usuário (*username*), e-mail e senha.

Como já temos uma conta criada, vou fazer o login na plataforma com meu usuário e senha. Para criar um projeto, vou clicar em "Pen". Vamos ser enviados para uma página com uma estrutura de três caixas de texto: HTML, CSS e JS.

O HTML, como o Paulo mostrou no Youtube, está relacionado à estrutura da página. O CSS a Rafa vai explicar para nós, e o Javascript é algo que vamos focar bastante durante a imersão. Rafa, o que podemos entender desses três e o que preparamos para as pessoas que estão fazendo a imersão?

Rafaella: HTML, CSS e JS são linguagens. O HTML é uma linguagem de marcação, então quando você abre uma página na internet, você vê títulos, parágrafos, imagens, e cada um desses elementos têm uma tag específica do HTML. Por exemplo, para adicionar um parágrafo usamos a tag `<p>`.

Note que, no HTML, as tags normalmente ficam entre os sinais de menor que e maior que (`<>`). Além disso, as tags fecham com `</>`, por exemplo `</p>`. Tudo aquilo que colocamos entre as tags de abertura e fechamento será considerado o conteúdo dessa tag. Vamos começar com o exemplo "Boas-vindas à Imersão Dev".

```
<p>Boas-vindas à Imersão Dev</p>
```

Guilherme: Repare que, para indicarmos que o parágrafo acabou, nós incluímos uma barra (|) ao fecharmos a tag.

Rafaella: Existem algumas tags que não abrem e fecham, porque às vezes você não coloca o conteúdo dentro delas, mas a maioria delas sim. Essa é uma tag exemplo para o parágrafo. Poderíamos ter também uma tag de título, que servirá de "header", como a `<h1>`, com o texto "Olá mundo".

```
<h1>Olá mundo</h1>
<p>Boas-vindas à Imersão Dev</p>
```

Perceba que na parte inferior da tela nós temos um campo que vai se atualizando conforme escrevemos nosso código, exibindo o resultado daquilo que programamos. Como utilizamos a tag `<h1>`, o texto "Olá mundo" ficou diferente, maior e em negrito, com mais cara de título, já que essa tag é específica para isso.

O CSS é um pouco diferente do HTML, é uma linguagem de estilização que serve, como o próprio nome diz, para estilizar os elementos do HTML, alterando cores, fontes, tamanhos de fonte e assim por diante. Para fazer isso, usamos o nome da tag seguido de abre e fecha chaves (`{}`), por exemplo `h1 {}`. Dentro das chaves, nós damos dois "Enter" para deixar o código mais organizado.

```
h1 {
}
```

Se eu quiser alinhar o texto para o centro da tela, por exemplo, posso usar a propriedade `text-align` com o valor `center`, finalizando com um ponto e vírgula. O CSS é bem intuitivo e usa várias palavras do inglês, então se você entende um pouco da língua vai ter um pouco de facilidade. Mas também não tem problema se não souber, você pode ir pesquisando e traduzindo para treinar.

```
h1 {
    text-align: center;
}
```

Feito isso, nosso "Olá mundo" será exibido no centro da tela. Também podemos alterar o tamanho da fonte com `font-size`, por exemplo para `100px` - uma medida de tamanho em pixels. Nosso `<h1>` vai ficar bem grande, e podemos ir alterando conforme nossas preferências. Nesse momento, também vamos incluir uma propriedade `color`: `red` para alterar a cor.


```
h1 {  
  text-align: center;  
  font-size: 100px;  
  color: red;  
}
```

Você pode explorar outras tags e estilizações para ir aprendendo um pouco mais. Agora que já vimos essas duas linguagens - lembrando que não são linguagens de programação, mas sim de marcação (HTML) e estilização (CSS), vamos conhecer o Javascript. Para isso, vamos abrir o projeto que faremos nessa aula.

Acho interessante vocês darem uma olhada em um artigo que escrevi sobre as diferenças entre essas três tecnologias usadas no front-end, para ter uma ideia sobre como funcionam, como surgiram e como utilizamos no dia a dia, inclusive com alguns exemplos de código.

[HTML, CSS e Javascript, quais as diferenças?](#)

Paulo: Antes de irmos para o Javascript, acho que você viu que o Gui e a Rafaella falam com muita propriedade, sabem as tags, sabem a sintaxe e assim por diante. Isso vem com o tempo, conforme você estuda esse mecanismo, assim como a sintaxe de uma língua. Eu não gosto tanto dessa comparação de língua falada com linguagem de programação, mas existem sim algumas analogias que podemos fazer.

É normal ter um choque inicial quando você vê essas tags, chaves, dois pontos e ponto e vírgula. Inclusive, assistindo ao vídeo, talvez seja até difícil distinguir onde é dois pontos e onde é ponto e vírgula. E você vai errar. E o que vai acontecer? Vai mostrar uma mensagem de erro ou, pior ainda, simplesmente não vai funcionar, não acontecerá nada.

Isso é normal, e você vai passar por esse tipo de situação aqui na Imersão Dev, nos seus estudos (seja na Alura ou em outro lugar) e no seu trabalho. Nós queremos que durante esses 10 dias você aprenda a solucionar mistérios, enfrentar problemas e ir além dos exercícios propostos.

Muitas das coisas que estamos passando aqui só vão ficar mais claras nas próximas aulas e conforme você vai praticando o assunto. É normal ficar confuso ou intrigado com alguns detalhes da sintaxe, como a inclusão ou não de ponto e vírgula, usar "Enter" entre as tags e assim por diante. Você é responsável por se colocar essas questões, experimentar e entender o que está acontecendo. Como dev, é importante ter essa curiosidade aguçada e fazer testes.

Algumas vezes você realmente ficará com dúvida, e nesses momentos é fundamental perguntar para os colegas e monitores no Discord para expandir e fixar seus conhecimentos.

Pelo que eu entendi, Rafa, vocês já trouxeram um pouco de HTML e CSS pronto para trabalharmos com Javascript.

Rafaella: Perfeito. Nós podemos brincar com Javascript tranquilamente, mas já temos um projeto preparado que será um pouco mais visualmente agradável, além de ter todo o tema que queremos ensinar. O que vamos fazer é simplesmente copiar e colar um pouco de código HTML e CSS.

Para copiar o código, você pode acessar minha conta no CodePen (<https://codepen.io/rafaballerini/pen/VwbxWdb>) e utilizar o botão "Fork" na parte inferior direita da página. Assim você consegue "bifurcar" o código e criar sua própria versão.

Código HTML

```
<html>
  <head>
    <title> Imersão Dev Alura - Aula 01</title>
  </head>
  <body>
    <div class="container">
      <h1 class="page-title">
        Calculadora de média</h1>
      <p class="page-subtitle">
        Descubra qual a sua média de nota de todas as provas!
      </p>
      
      </div>
      <a href="https://alura.com.br/" target="_blank">
        
        </a>
      </body>
</html>
```

Código CSS

```
body {
  font-family: 'Roboto Mono', monospace;
  min-height: 400px;
  background-image:
url('https://image.flaticon.com/icons/png/512/201/201635.png');
  background-color: #000000;
  background-size: 80vh;
  background-position: center bottom;
  background-repeat: no-repeat;
}

.container {
  text-align: center;
  padding: 20px;
  height: 100vh;
}

.page-title {
  color: #ffffff;
  margin: 0 0 5px;
}

.page-subtitle {
  color: #ffffff;
  margin-top: 5px;
}

.page-logo {
  width: 200px;
}

.alura-logo {
  width: 40px;
  position: absolute;
  top: 10px;
  right: 10px;
}
```

Paulo: Para a maioria das aulas nós já vamos trazer um HTML e CSS pronto, pois o mais importante é o Javascript, que vamos começar do zero. Repare que no código que

disponibilizamos existem vários conteúdos que não comentamos, mas isso também é prática comum na programação.

Para nossa primeira aula, na qual calcularemos uma média, nós vamos escrever toda a funcionalidade, mas a parte do layout já está preparada. E você pode ir além, dá para alterar essa página inteira fazendo ajustes no HTML e CSS. Por enquanto, nosso foco será totalmente no Javascript.

Rafaella: Exato, vamos focar no Javascript para explorarmos toda a parte da lógica, mas vocês também vão ver HTML e CSS. Em relação a isso, acho que é interessante, depois da aula, você tentar entender o que são essas tags, o que estamos estilizando e até mesmo testar algumas estilizações para ver como a página vai reagir.

Gui, eu acho que talvez seja legal mudarmos o layout do CodePen, trazendo o código para à esquerda, para vermos a página inteira. No canto superior direito, temos um botão "Change View" que permite mudar o layout da página. Usaremos a opção "Editor view".

Se você quiser, também pode puxar as caixas de texto do HTML e CSS para cima, minimizando-as, assim conseguiremos focar no Javascript. Gui, você pode começar a escrever o nosso código Javascript,

Guilherme: A ideia é criarmos uma calculadora de média. Um computador interpreta os comandos que a gente insere, mas é necessário armazenar na memória dele algumas informações. Uma forma de fazermos isso é através do que chamamos de variáveis. Quando criamos uma variável, armazenamos na memória do computador um determinado valor.

Por exemplo, eu quero que o computador guarde o meu nome, "Guilherme". Se eu simplesmente escrever Guilherme na caixa de texto do Javascript, nada vai acontecer. O código vai ser executado automaticamente, algo que vem ativado por padrão no CodePen. Se acessarmos as configurações (*Settings*), veremos que o salvamento automático está ligado (*save automatically?*), assim como a execução automática (*auto-updating preview*). Vamos desabilitar ambos.

Paulo: Em "Pen Details", além de ser possível dar um nome e uma descrição para o seu projeto, é importante incluir nas tags "immersaodev". Assim, nós conseguiremos puxar o seu projeto e mostrar para outras pessoas o que está sendo desenvolvido. Você também pode incluir outras tags se quiser.

Guilherme: Após salvar as configurações, teremos um botão de "Run" na parte superior direita da tela, que nos permite executar o código.

Paulo: A gente prefere manter dessa forma pois evitamos que o CodePen acuse erros a todo momento, o que poderia atrapalhar a explicação, mas você pode configurar da maneira que preferir.

Guilherme: Note que eu digitei somente meu nome, Guilherme, e nada aconteceu, pois o Javascript não sabe o que é "Guilherme". Além disso, na parte inferior da caixa de texto, temos um ponto de exclamação em vermelho indicando que algo não está certo. Lembra que o Paulo comentou do console no navegador? Nós também temos um console no CodePen, no canto inferior esquerdo da página.

Quando clicamos em "Console", ele mostra alguns detalhes - nesse caso, um erro *"Guilherme is not defined"*. Para definir que "Guilherme" é meu nome, vou criar uma variável nome recebendo o valor `Guilherme`.

```
nome = Guilherme
```

Porém, ao executar, continuaremos recebendo o mesmo erro. Existem algumas coisas que precisamos passar para o computador, mais especificamente na linguagem Javascript, mas também em muitas outras. Sempre que vamos tratar de uma palavra ou texto no Javascript, utilizamos um par de aspas duplas (`" "`).

```
nome = "Guilherme"
```

Ao executar, não acontecerá nada. A ideia é que o texto "Guilherme" fique reservado em um espaço da memória do computador que é acessado a partir da palavra nome. Como fazemos para deixar isso mais claro?

Rafaella: Nós usamos a palavra-chave `var`, de "variável".

```
var nome = "Guilherme"
```

Guilherme: Repare que a cor de `var nome` mudou. Isso porque a maioria das palavras-reservadas do Javascript tem uma cor em vermelho, variáveis em laranja e strings (textos) em amarelo, e assim por diante - isso no modo padrão de cores do CodePen. Agora eu quero exibir esse nome no console. Como fazemos isso?

Rafaella: Como o Gui falou, nós só estamos armazenando o valor em uma variável, não estamos imprimindo. Nós utilizamos o console para verificar o que está acontecendo em nosso programa. No dia a dia, não pedimos para os usuários digitarem comandos no console ou coisa do tipo. Normalmente usamos o console para imprimir algum valor, por exemplo, facilitando que quem está desenvolvendo consiga entender o que está acontecendo no programa.

Vamos imprimir nossa primeira variável para visualizarmos o valor dela no momento. Para isso, usaremos a função `console.log()`. Inicialmente, passaremos um texto entre aspas, por exemplo "Bem-vindo". Depois das aspas, incluiremos o sinal de positivo (`+`) e a nossa variável `nome`.

```
console.log("Bem-vindo" + nome)
```

Guilherme: Ao clicar em "Run", vai aparecer no console "Bem-vindoGuilherme", tudo junto.

Rafaella: Isso a gente arruma colocando um espaço antes das aspas.

```
console.log("Bem-vindo " + nome)
```

Rafaella: Se você quiser, pode criar uma variável `boasVindas` com o texto "Bem-vindo", por exemplo. Nesse caso, como é algo que sempre vai aparecer quando fizermos o `console.log()`, não criaremos uma variável só para isso.

Guilherme: Olha que legal, Rafa. Quando passamos a variável `nome`, ele sabe que precisa mostrar o valor `Guilherme`. Já se colocarmos "nome", entre aspas duplas, o resultado será "Bem-vindo nome", e não o conteúdo dessa variável.

Rafaella: Exatamente. Agora podemos criar variáveis para começarmos a brincar com as notas da escola ou faculdade.

Paulo: Essa brincadeira que o Gui fez de colocar o nome entre aspas é uma dúvida que os alunos e alunas podem ter. E se eu não usar o sinal de positivo ou não usar aspas no "Bem-vindo"? E se eu não usasse os parênteses no `console.log()`? A gente poderia passar 10 aulas só falando do `console.log()` ou de variáveis do tipo texto, que chamamos de "strings".

São vários detalhes que estão acontecendo aqui, e o nível de profundidade que vamos abordar em cada um desses casos não será muito grande, afinal queremos programar e desenvolver. Porém, se essa vai ser a sua carreira, você tem que conhecer a fundo cada um desses conceitos.

No começo é normal se sentir inseguro ou insegura em relação ao que vai acontecer caso você experimente com o código. Mas você deve levar isso como um aprendizado, com aquela sensação de exploração que vai te impulsionar a novos conhecimentos.

Rafaella: Exatamente. Agora podemos criar a nossa segunda variável. Uma coisa que eu acho interessante vocês saberem é que no Javascript, diferente de outras linguagens, normalmente utilizamos um padrão chamado "camel case" para escrever os nomes das variáveis. Nesse padrão, o primeiro nome começa com a letra minúscula e o segundo (e os seguintes) com letra maiúscula.

Por exemplo, vamos criar a variável `notaDoPrimeiroBimestre`. Repare que cada palavra, exceto a primeira, se inicia com letra maiúscula.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre

console.log("Bem-vindo " + nome)
```

Rafaella: Se você colocar em letra minúscula também vai funcionar, mas é um padrão bem utilizado no Javascript. Vamos passar um valor para essa nota, por exemplo `9`.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = "9"

console.log("Bem-vindo " + nome)
```

Rafaella: E agora podemos imprimir essa primeira nota para verificarmos se está funcionando.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = "9"

console.log("Bem-vindo " + nome)
console.log(notaDoPrimeiroBimestre)
```

Paulo: Repare que enquanto o Gui digitava, apareceu uma sugestão de autocompletar na tela. Muitos editores, como o CodePen e o Visual Studio Code, têm esse tipo de facilidade, com sugestões de autocompletar ou mesmo de correção de bugs. Com o tempo, utilizando essas ferramentas, você ganha fluência na linguagem.

Guilherme: Executando mais uma vez, ele vai exibir o "Bem-vindo Guilherme" e a nota "9". Rafa, vamos começar a fazer algumas operações antes de calcularmos nossa

média de fato. Vou criar uma variável `notaFinal` com o padrão **camel case**, e ela vai receber a `notaDoPrimeiroBimestre` somada a mais um ponto, como se fosse um ponto de presença. Para isso, posso usar o sinal de soma (+) e o valor `1`. Além disso, vamos imprimir essa `notaFinal`.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = "9"
var notaFinal = notaDoPrimeiroBimestre + 1

console.log("Bem-vindo " + nome)
console.log(notaFinal)
```

Guilherme: A ideia é que a nota final "10" seja mostrada. Porém, quando executarmos, será exibido "91" no console. Acho que não era isso que a gente queria.

Rafaella: Por que isso aconteceu, Gui?

Guilherme: Da mesma forma que sabemos o que é um número e o que é uma palavra, precisamos informar ao computador quando estamos trabalhando com um número e quando estamos trabalhando com uma palavra. Sempre que trabalharmos com números, não utilizaremos as aspas duplas, deixando o valor sozinho.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = 9
var notaFinal = notaDoPrimeiroBimestre + 1

console.log("Bem-vindo " + nome)
console.log(notaFinal)
```

Repare que a cor dele até fica diferente da cor das outras palavras. Em computação, chamamos palavras e textos de "strings", e são colocados em aspas. Quando queremos utilizar valores para fazer operações, não usamos as aspas. Em outras linguagens como o Java, esse tipo de coisa que fizemos - tentar "somar" uma palavra com um número - geraria um erro. Em nosso caso, nós concatenamos (ou seja, juntamos) as duas strings, gerando o valor `91`.

Corrigido nosso erro, ao executarmos novamente teremos o valor `10`, que é o comportamento esperado. Agora sabemos como indicar para o Javascript o que são palavras ou valores numéricos.

Paulo: Como o Guilherme falou, em outras linguagens esse tipo de cenário traria resultados diferentes - inclusive somar duas strings numéricas poderia resultar no valor

correto. Isso depende da forma como a linguagem interpreta o seu código, e isso às vezes incorre em bugs e erros inesperados.

Repare que não tivemos nenhum erro em nosso código, ele simplesmente executou e imprimiu o valor, mas não era o esperado. Para nós, seres humanos, foi um erro, mas para a máquina tudo foi interpretado corretamente. Nós vamos passar por muitos casos como esse durante a imersão.

Rafaella: Agora já temos a `notaDoPrimeiroBimestre`, e agora queremos fazer a nota anual, abrangendo os quatro bimestres. Vamos criar algumas variáveis para isso, `notaDoSegundoBimestre`, `notaDoTerceiroBimestre` e `notaDoQuartoBimestre`, com os valores `7`, `4` e `2`, respectivamente.

Nossa `notaFinal` agora passará a ser a soma das notas desses quatro bimestres dividida por quatro. Na programação, a divisão utiliza o símbolo `/` (barra), enquanto a multiplicação utiliza o `*` (asterisco).

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = 9
var notaDoSegundoBimestre = 7
var notaDoTerceiroBimestre = 4
var notaDoQuartoBimestre = 2

var notaFinal = notaDoPrimeiroBimestre + notaDoSegundoBimestre +
notaDoTerceiroBimestre + notaDoQuartoBimestre / 4

console.log("Bem-vindo " + nome)
console.log(notaFinal)
```

Guilherme: Porém, quando executarmos, teremos um resultado estranho: `20.5`. Não era isso, né?

Rafaella: Não mesmo. Uma coisa que vocês provavelmente já viram nas aulas de matemática é que, quando queremos fazer uma operação antes de outra, utilizamos os parênteses. Da maneira como construímos o cálculo, ele primeiro divide `notaDoQuartoBimestre` por `4`, para então fazer a soma dos outros valores - afinal, a divisão é uma operação prioritária em relação à soma.

Queremos primeiramente somar todos os valores para então dividi-los, resultando na média. Para isso, usaremos os parênteses.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = 9
var notaDoSegundoBimestre = 7
var notaDoTerceiroBimestre = 4
var notaDoQuartoBimestre = 2

var notaFinal = (notaDoPrimeiroBimestre + notaDoSegundoBimestre +
notaDoTerceiroBimestre + notaDoQuartoBimestre) / 4

console.log("Bem-vindo " + nome)
console.log(notaFinal)
```

Paulo: É uma boa prática utilizar os parênteses, mesmo que você conheça a precedência dos operadores, por uma questão de legibilidade. Assim, outras pessoas conseguirão ler e entender o código mais tranquilamente.

Guilherme: Executando novamente, teremos o valor **5.5**.

Rafaella: Perfeito. Vimos que é possível fazer essa soma e divisão com números inteiros. Mas e se tivéssemos números quebrados? Em situações como essa, poderíamos ter resultados com vários números decimais, como **5.5432110**. Vamos ver um exemplo.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = 9.4355
var notaDoSegundoBimestre = 7.234
var notaDoTerceiroBimestre = 4.234
var notaDoQuartoBimestre = 2.183

var notaFinal = (notaDoPrimeiroBimestre + notaDoSegundoBimestre +
notaDoTerceiroBimestre + notaDoQuartoBimestre) / 4

console.log("Bem-vindo " + nome)
console.log(notaFinal)
```

Rafaella: Como resultado, teremos `5.771625`. Isso pode acontecer, às vezes pedimos para o usuário inserir um número inteiro em nosso programa e ele acaba não fazendo isso. Pensando nisso, quando desenvolvemos programas nós tentamos prevenir que ao menos o resultado esteja certo, formatado da maneira como desejamos - por exemplo, arredondando para uma ou duas casas decimais. Assim, conseguimos entregar para o usuário uma experiência melhor.

Para isso, podemos usar uma função semelhante ao `console.log()`, chamada `toFixed()`, que consegue ajustar o número de casas decimais do nosso valor. Vamos criar uma nova variável, `notaFixada`, que receberá `notaFinal.toFixed()`, e entre os parênteses passaremos o número de casas decimais que desejamos fixar - nesse caso, `1`. Agora passaremos a imprimir a `notaFixada`, e não mais a `notaFinal`.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = 9.4355
var notaDoSegundoBimestre = 7.234
var notaDoTerceiroBimestre = 4.234
var notaDoQuartoBimestre = 2.183

var notaFinal = (notaDoPrimeiroBimestre + notaDoSegundoBimestre +
notaDoTerceiroBimestre + notaDoQuartoBimestre) / 4

var notaFixada = notaFinal.toFixed(1)

console.log("Bem-vindo " + nome)
console.log(notaFixada)
```

Guilherme: Ao executar, teremos como resultado `5.8`.

Rafaella: Perfeito, agora conseguimos fazer com que o próprio programa arredonde a nota.

Guilherme: Interessante essa parte do `toFixed()`. Não fui, nem o Paulo e nem a Rafa que criou essa função, mas estamos utilizando em nosso programa. Isso vai ser algo muito comum: vamos utilizar soluções que fazem parte do escopo da linguagem para criarmos nossos programas.

Quando pegamos um determinado valor numérico e passamos para o `toFixed()`, ele faz um monte de coisas por trás que a gente nem sabe quais são. Se passarmos `2` casas decimais para essa função, ela terá outro comportamento, mostrando o resultado `5.77`. Se não colocarmos nenhum valor, representando nenhuma casa decimal, o resultado será `6` - ou seja, a função está arredondando para cima.

Repare que nós não trouxemos um trecho de código e colocou em nosso projeto para termos esse comportamento, o que na verdade é algo que também pode acontecer. Estamos utilizando funções do próprio Javascript para resolver determinados problemas nossos.

Rafaella: Acho que agora podemos fazer uma revisão. Já conseguimos construir nossa calculadora de média, e agora vamos rever os conceitos que aprendemos nessa aula. Antes disso, vamos voltar para os valores originais.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = 9
var notaDoSegundoBimestre = 7
var notaDoTerceiroBimestre = 4
var notaDoQuartoBimestre = 2

var notaFixada = notaFinal.toFixed(1)

var notaFinal = (notaDoPrimeiroBimestre + notaDoSegundoBimestre +
notaDoTerceiroBimestre + notaDoQuartoBimestre) / 4

console.log("Bem-vindo " + nome)
console.log(notaFixada)
```

Rafaella: Para fazermos nossa revisão, existe mais um conceito interessante de ensinar a vocês, que são os comentários no código. Cada linguagem tem um formato para comentários - algumas usam asteriscos, outras parênteses e assim por diante. No Javascript nós utilizamos duas barras (`//`).

Tudo que for escrito no código depois de duas barras não será executado pelo interpretador da linguagem, permitindo a adição de textos ao longo do projeto.

```
var nome = "Guilherme"
var notaDoPrimeiroBimestre = 9
var notaDoSegundoBimestre = 7
var notaDoTerceiroBimestre = 4
var notaDoQuartoBimestre = 2
```

```
var notaFixada = notaFinal.toFixed(1)

var notaFinal = (notaDoPrimeiroBimestre + notaDoSegundoBimestre +
notaDoTerceiroBimestre + notaDoQuartoBimestre) / 4

console.log("Bem-vindo " + nome)
console.log(notaFixada)

//isso é um comentário
```

Rafaella: Se tivéssemos escrito "isso é um comentário" sem as duas barras, teríamos um erro "unexpected identifier" no console. Utilizando os comentários, podemos escrever a revisão dos conceitos desta aula.

O primeiro conceito que conhecemos foram as variáveis, espaços na memória do computador que utilizamos para referenciar valores.

Guilherme: Também vimos as strings, que são as palavras e textos que colocamos no código usando as aspas.

Rafaella: Vimos como funciona o `console.log()` para imprimir no console.

Guilherme: E também o `toFixed()` para ajustar as casas decimais das nossas notas.

Rafaella: Vimos as operações matemáticas, soma, divisão, multiplicação e subtração, e a concatenação, por exemplo quando nós juntamos uma string e uma variável em nosso `console.log()`. Além disso, utilizamos ao longo de toda a aula o console para imprimir os nossos resultados.

Agora, vamos colocar alguns desafios para o pessoal de casa.

Guilherme: Eu tenho um desafio. A gente já somou todos os valores das notas e dividiu por quatro. Eu poderia quebrar isso em duas etapas, primeiro somando os valores e depois fazendo a divisão do resultado dela. Eu também poderia incluir notas com duas casas decimais ou verificar se o aluno foi aprovado ou não de ano dependendo da nota.

Paulo: Esse precisa de alguns comandos a mais que não vimos ainda.

Rafaella: Com certeza. Esse é bem interessante, e se você já souber como funciona, pode fazer. Um desafio que eu gosto de propor é que você estilize esse fundo. Nós colocamos uma imagem que está dentro do HTML, e eu desafio você a trocar essa imagem, trocar a cor do fundo e fazer outras alterações de estilização nessa página.

E outro desafio que eu gostaria de propor, um pouco mais avançado, é que você tente escrever na página o que estamos calculando no console. Se você quiser pesquisar como fazer isso e tentar fazer, vai ser bem legal.

Por fim, tem também um desafio muito comum quando estamos estudando na faculdade, que é um conversor de temperaturas. Ao invés de você calcular a média das notas, calcular quanto determinada temperatura em graus Celsius é em graus Fahrenheit.

Paulo: Também poderia ser de milha para quilômetro...

Rafaella: Ou de moeda também, você pode entrar com real e sair com dólar.

Paulo: Eu vou deixar de desafio o contrário do que o Guilherme colocou. A gente tem esse `var notaFinal` que já soma e divide, e ele pediu para você quebrar em duas etapas. Meu desafio é que você coloque todo o cálculo no `console.log()`, incluindo a soma, a divisão e o `toFixed()`. Vai ter uma brincadeira de sintaxe aí que pode atrapalhar a sua cabeça.

Lembrando que esses desafios não significam que essa é a forma ideal de escrever código, e muitas vezes o código pode até ficar menos legível. Eu acho que o nível em que criamos agora está muito bom, com os passos bem inteligíveis, e ficar quebrando em muitos passos ou resumindo demais pode ficar confuso.

Assim que você terminar essa aula, salve e pegue o link que você recebeu, que é referente ao seu projeto. Mande ele no Whatsapp da família, no Discord para tirar dúvidas e para seus colegas. Você também pode tirar um screenshot, colocar em seu instagram e mandar para nós.

Você pode até gravar sua tela utilizando plugins do Chrome e fazer um vídeo explicando com suas próprias palavras. Isso ajuda a fixar os seus conhecimentos.

Quero ver o seu projeto no ar, você compartilhando no Discord. E lembre-se de indicar um amigo ou amiga para participar da Imersão Dev, que ainda dá tempo.

Rafaella: Acho que também é interessante vocês entenderem que estamos fazendo vários projetos nessa imersão, e todos eles servem para o seu portfólio. É o seu portfólio que mostra que você tem conhecimento sobre um assunto, portanto salve esses links do CodePen, que ao final dessa imersão descobriremos onde colocá-los. Vai ser uma surpresa muito legal.

Paulo: Então a gente se vê amanhã, foco e código. Tchau!

Rafaella: Tchau, tchau!

Guilherme: Tchau, pessoal!