



Frontend

Workshop – Local Hack Day

Florestal

Universidade Federal de Viçosa





Sobre nós

● Gustavo Viegas

- 6 anos de experiência profissional.
- Atualmente é prestador de serviços como PJ.
- Graduando em Ciência da Computação na UFV – CAF desde 2017.

● Bruno Marra

- 3 anos de experiência profissional
- Técnico em Informática para Internet pelo CEFET-MG
- Vencedor da 26ª META também pelo CEFET-MG com o projeto Newpreço
- Graduando em Ciência da Computação na UFV – CAF desde 2017.





Conteúdo do Curso

1. Como funciona a Web.
2. Marcação.
3. Estilização.
4. Interatividade.
5. Integração com Backend.
6. Frameworks e Componentização.





Conteúdo do Curso

- Desafios práticos.
 - Utilização do Codepen e Sandbox.
 - As URLs estão disponíveis em
- Jogos.
- Um simples projeto final.





O que é Frontend?





O que é Frontend?

- Frontend é a camada de apresentação que os usuários interagem com um sistema.
 - Pode ser uma página web, um aplicativo, um programa desktop, etc.
- Executado diretamente no dispositivo do usuário.





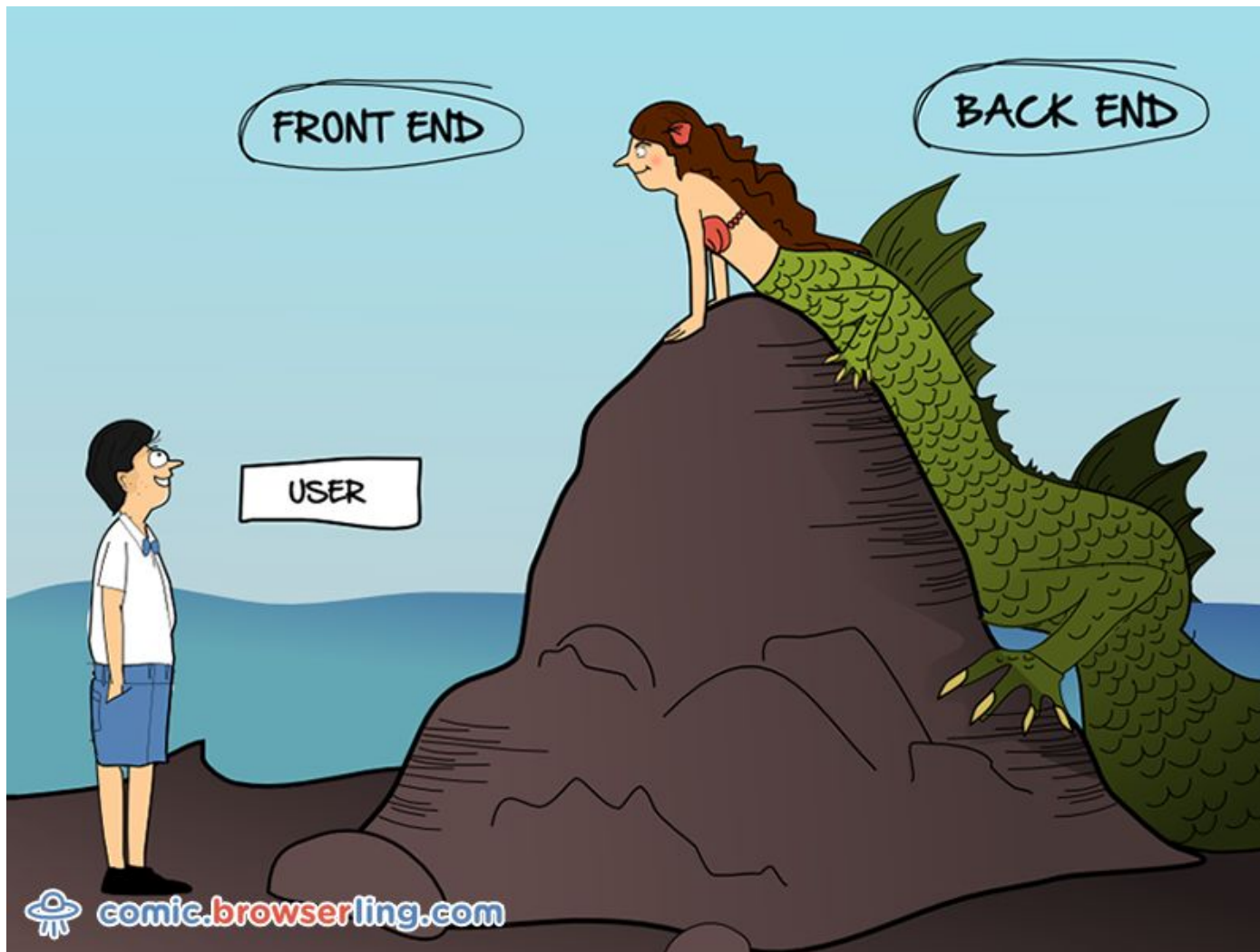
O que é Backend?

- Backend é a camada de recuperação e tratamento de dados e negócio que os usuários não possuem conhecimento ou visualização.
 - Recebe e retorna dados através de protocolos.
- Não necessariamente executa no dispositivo do usuário.
 - Na maioria das vezes, não executa.





Como a Web Funciona





Como a Web Funciona

- O Frontend na web é construído em browsers.
 - Conhecem apenas a linguagem de marcação **HTML**, de estilização **CSS** e de programação **JavaScript**.
- O Backend na web é construído em servidores.
 - Sem restrições de tecnologias, desde que os protocolos sejam implementados corretamente.





Como a Web Funciona

1. Você solicita o browser uma qual site quer ver.
2. O browser procura o site em um servidor DNS.
3. O browser solicita o servidor para enviar uma página específica.
4. O servidor envia o arquivo HTML para o browser.
5. O browser começa a ler o HTML do início ao fim do arquivo, parando para carregar recursos adicionais:
 - Fontes, Imagens e Arquivos CSS e JS.
6. O browser faz requisições para recursos adicionais.
7. Quando o browser chega ao fim do arquivo, ele começa a renderizar e mostrar a página.





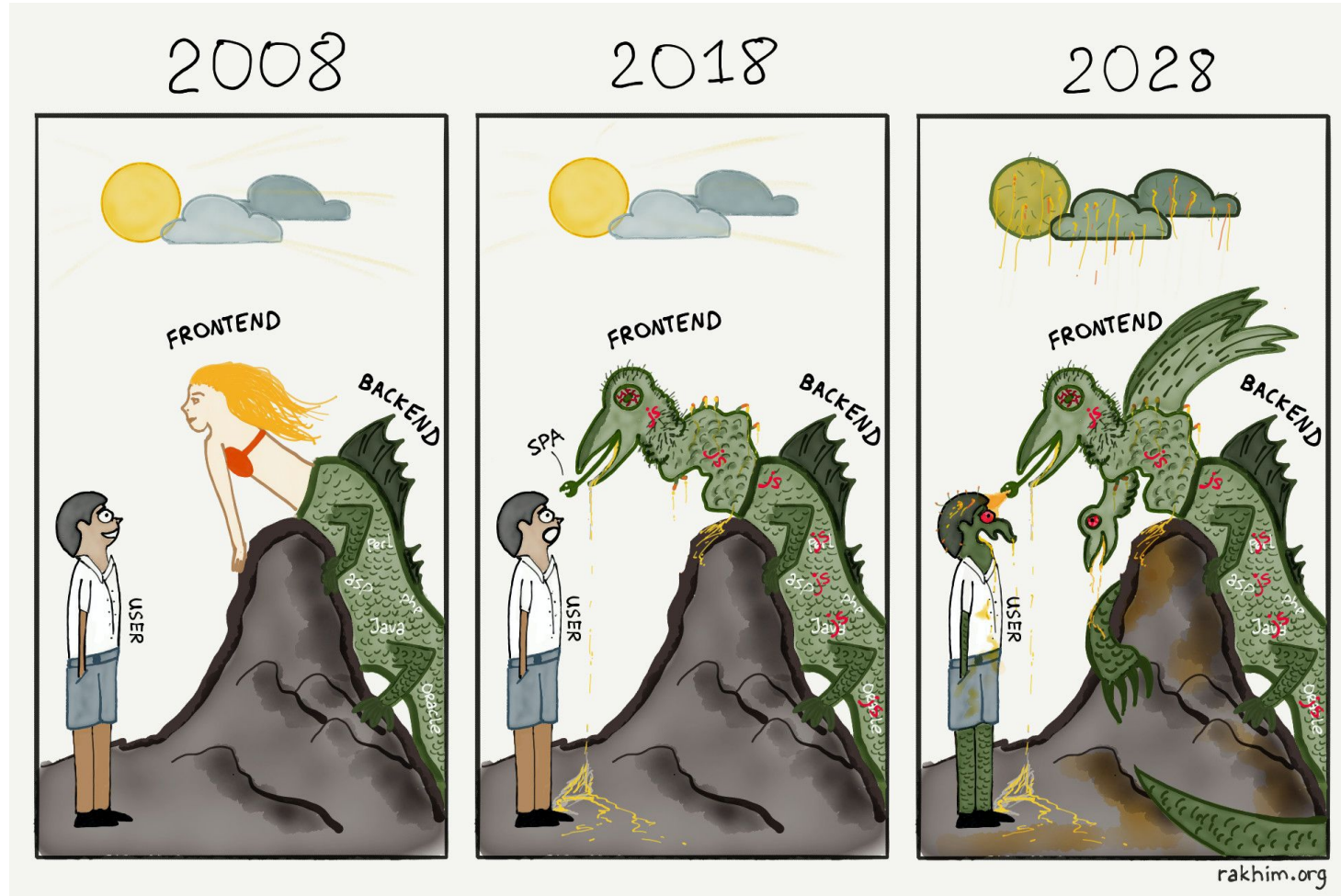
Como a Web Funciona

All HTML CSS JS XHR Fonts Images Media WS Other							
Status	Domain	File	Type	Size	0 ms	320 ms	640 ms ^
200	developer.mozilla.org	/en-US/	html	54.47 KB	13 ms		
200	developer.mozilla.org	mdn.9e3ae837b3dd.css	css	79.63 KB	13 ms		
200	developer.mozilla.org	home.6909465c3e7e.css	css	12.33 KB	17 ms		
200	developer.mozilla.org	locale-en-US.520ecdcaef8c.css	css	524 B	228 ms		
200	developer.mozilla.org	javascript.414b87adc480.js	js	3.14 KB	17 ms		
200	developer.mozilla.org	main.0ed22a584ec2.js	js	158.75 KB	16 ms		
200	developer.mozilla.org	newsletter.a85c5c2892e5.js	js	3.83 KB	223 ms		
200	developer.mozilla.org	web-docs-sprite.22a6a085cf14.svg	svg	10.09 KB	13 ms		
200	developer.mozilla.org	hero-dino.6c51deebd4af.png	png	10.99 KB	18 ms		
200	developer.mozilla.org	grid-tools-bg.33d7e775dcb5.svg	svg	99.85 KB	19 ms		
200	developer.mozilla.org	ZillaSlab-Regular.subset.bbc33fb47cf6.woff2	woff2	33.02 KB	20 ms		
200	developer.mozilla.org	ZillaSlab-Bold.subset.e96c15f68c68.woff2	woff2	33.27 KB	22 ms		
200	www.google-analytics.com	analytics.js	js	43.10 KB	23 ms		
200	developer.mozilla.org	favicon144.e7e21ca263ca.png	png	1.30 KB	15 ms		
200	developer.mozilla.org	favicon32.7f3da72dcea1.png	png	441 B	229 ms		
200	www.google-analytics.com	collect?v=1&_v=j73&aip=1&a=676378077&t=pa...	gif	35 B	16 ms		





Como a Web Funciona





Marcação

- HTML (**H**yper**T**ext **M**arkup **L**anguage)
 - Linguagem de marcação que o navegador interpreta para exibição de conteúdo.
- Cada elemento visual de uma página web consiste de uma ou mais tags.
- Websites possuem centenas de tags, na maioria das vezes aninhadas.
 - A página inicial do G1 possui mais de 4500 tags!





Marcação

- Existem diversas tags com os mais diversos propósitos.
 - Ex: parágrafos, tabelas, links, imagens, rodapés, etc.
- Um elemento HTML consiste de um conteúdo encapsulado por uma tag de abertura e uma tag de fechamento.



Marcação

```
<p> O time mais legal da NBA é o Los Angeles Clippers! </p>
```

Tag de abertura

Conteúdo

Tag de fechamento

- Tag de abertura
 - Nome do elemento (p) envolvido entre colchetes angulares de abertura (<) e fechamento (>). Indica onde o elemento começa ou inicia o efeito.
- Conteúdo: pode ser um texto, ou outro elemento.
- Tag de fechamento
 - O mesmo de abertura, mas tem um / antes do nome do elemento. Indica o fim do elemento.

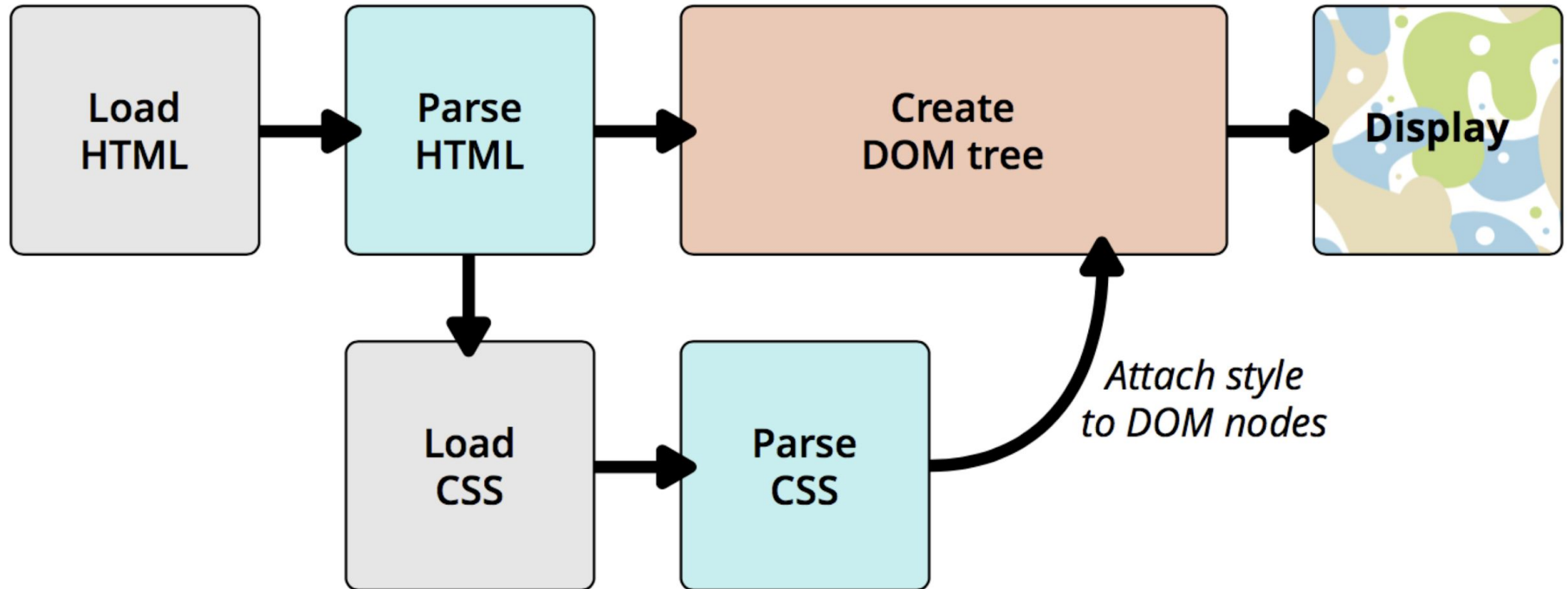
Marcação

```
  
<meta name="charset" content="utf-8">  
<link rel="stylesheet" href="./css/master.css">
```

- Existem tags especiais que já exibem algum elemento visual ou não possuem conteúdo dentro delas
 - Ex: Imagens (img), Inputs (input), Meta Tags, Source de CSS, etc.
- Atributos adicionam peculiaridades e nos permite identificar e customizar cada tag.
 - Adicionadas com a sintaxe **chave="valor"**
 - No exemplo acima, src, name, content, rel e href são atributos de suas tags.



HTML



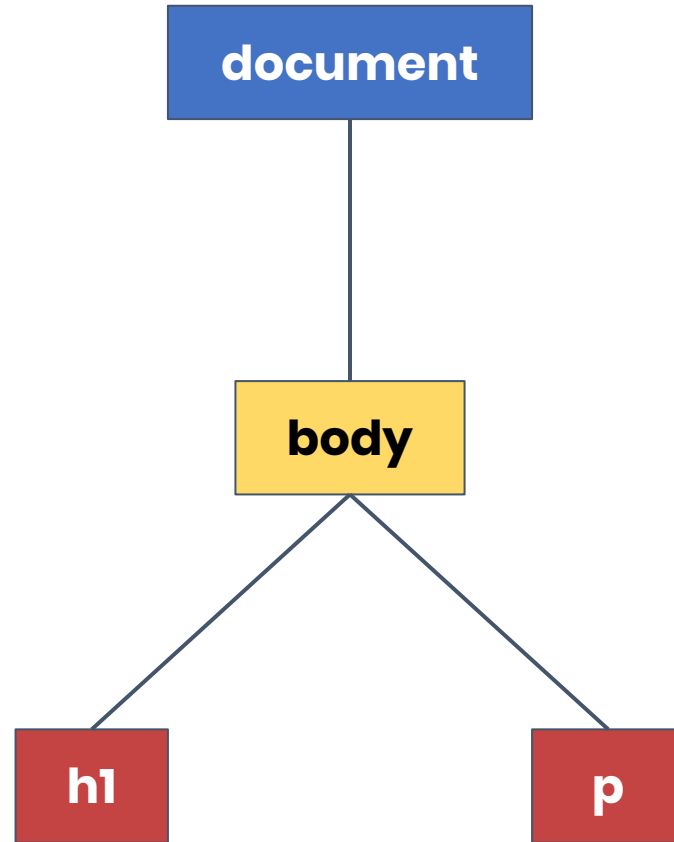
HTML

```
<!DOCTYPE html>
<html>
  <!-- Cabeçalho -->
  <head>
    <title>Título da página</title>
    <meta charset="utf-8">
  </head>

  <!-- Conteúdo da página -->
  <body>
    <h1>Hello world!</h1>
    <p>Página simples que só possui um título e um parágrafo.</p>
  </body>
</html>
```



HTML





Demo

- <https://codepen.io/brunomarram/pen/OJPyWmv>





Desafio #1





Desafio #1

- <https://github.com/brunomarram/hackday-2019>

Atividade 1 – HTML





Estilização

- Ótimo, agora eu já consigo fazer um site.
 - Porém horrível e estático.
- Vamos melhorar o estilo primeiro, é a função dos **C**ascading **S**tyle **S**heets.
- A maioria das tags HTML já possuem estilos definidos por padrão (como você viu no exemplo).
 - Eles podem ser substituídos, em cascata.





Estilização

- O que podemos mudar?
 - a. Tipografia (fontes, tamanho, estilo da fonte, etc).
 - b. Cores.
 - c. Bordas, quinas, decorações.
 - d. Layout.
 - e. Posições.
 - f. Espaçamentos, margens.
 - g. Animações.
 - h. ...
 - i. e muito mais coisas!



Estilização

- CSS pode ser aplicado em tags HTML de três maneiras:
 - Inline com o atributo style em uma tag;
 - `<div style="background: blue; color: white;"> Oi </div>`
 - Através de uma tag `<style>` na página HTML.
 - `<style> div { background: blue; color: white; } </style>`
 - Através de um arquivo CSS externo.
 - `<link rel="stylesheet" href="/arquivo.css" />`



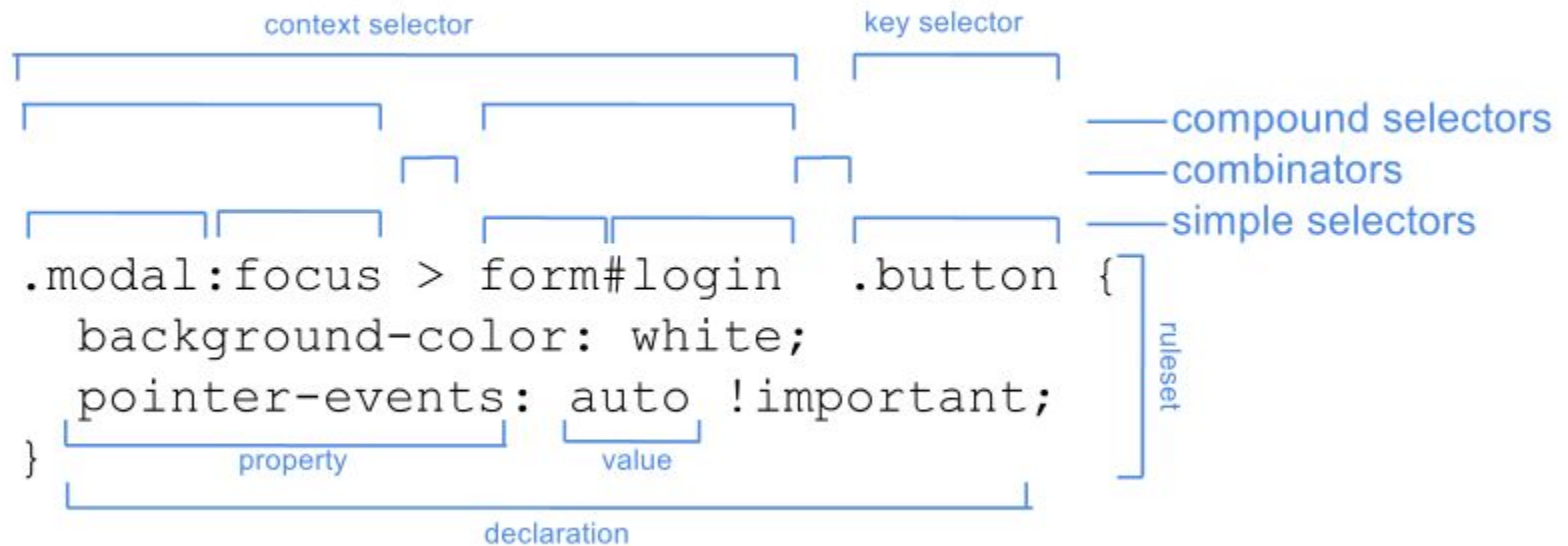
Estilização

- Para aplicar estilos em elementos através da tag style ou arquivo externo usamos de seletores.
- Uma mesma regra pode ser aplicada para vários elementos, ou para todos menos um elemento.
- A linguagem de seletor é rica e possui diversos operadores para auxiliar o desenvolvedor.
 - O mais comum é o uso de seletores de **CLASSES** e **IDS** (atributos class e id de qualquer tag).





Estilização





Demo

- <https://codepen.io/brunomarram/pen/OJPyWmv>





Desafio #2





Desafio #2

- <https://github.com/brunomarram/hackday-2019>

Atividade 2 – CSS





Dinamização

- Ótimo, agora eu já consigo fazer um site bonito.
 - Porém estático.
- Vamos torná-lo dinâmico com o uso de **JavaScript**.
- JS é uma linguagem de programação que o browser sabe como interpretar.
 - Além disso, o browser oferece uma porção de APIs para interagir com o DOM.
 - Dessa forma podemos alterar CSS e HTML mediante algum evento, ou quando quisermos.





O JS





O JS

- Mas o que posso fazer com JS:
 - conteúdos atualizados dinamicamente
 - controlar multimídias
 - imagens animadas
 - e mais um mundo de outras coisas



O JS

**E se linguagens
de programação
fossem
cachorros?**



O JS

O Javascript seria um vira-lata: É um super remendo de um monte de coisas diferentes, consegue comer ração e pomba e tem muita gente que adora



JS



O JS

```
1 | var para = document.querySelector('p');  
2 |  
3 | para.addEventListener('click', atualizarNome);  
4 |  
5 | function atualizarNome() {  
6 |     var nome = prompt('Insira um novo nome');  
7 |     para.textContent = 'Jogador 1: ' + nome;  
8 | }
```





O JS

- Para utilizar JavaScript no frontend, basta declarar o código dentro das tags .
- Também se pode carregar um arquivo externo com extensão .js pelo atributo src. Exemplo:
`<script src="./index.js"></script>`
- Normalmente declarado no fim do body.
- Output no console do modo desenvolvedor.



Tipos de dados

- Tipos de dados:
 - Boolean: true/false.
 - Nulo: null.
 - Números: Number.
 - Strings: String.
 - Objetos: Object.
- Lembrete: dinamicamente tipada!



Dinamização





Objetos

- Linguagem essencialmente orientada a objetos.
- Existem diversos objetos úteis com propriedades e métodos convenientes. Exemplos: qualquer array.
- Acesso às propriedades com a mesma sintaxe de C++. `objeto.propriedade`
- Aninhamento de objetos e propriedades: `objeto.metodo().propriedade.metodo()`
- Mas Arrays podem ser acessados com seu índice. `array[90]`



Objetos

```
const x = 90

function fazAlgo() {
  console.log(y) // undefined
  let y = "Teste"
  x = {legal: true} // Gera exceção
}

let fazAlgo = parseFloat("900.233") // Também gera exceção

const getId = function (n) {
  return Math.sqrt(n ** 3)
}

const animais = ["Leão", , "Lobo", "Peixe", , "Macaco"]

// Misturando tudo!
const dados = {
  animais: animais,
  id: getId(0),
  nome: "Teste"
}
```





Demo

- <https://codepen.io/brunomarram/pen/WNbQRKm>





Desafio #3





Desafio #2

- <https://github.com/brunomarram/hackday-2019>

Atividade 3 – JS



JSON

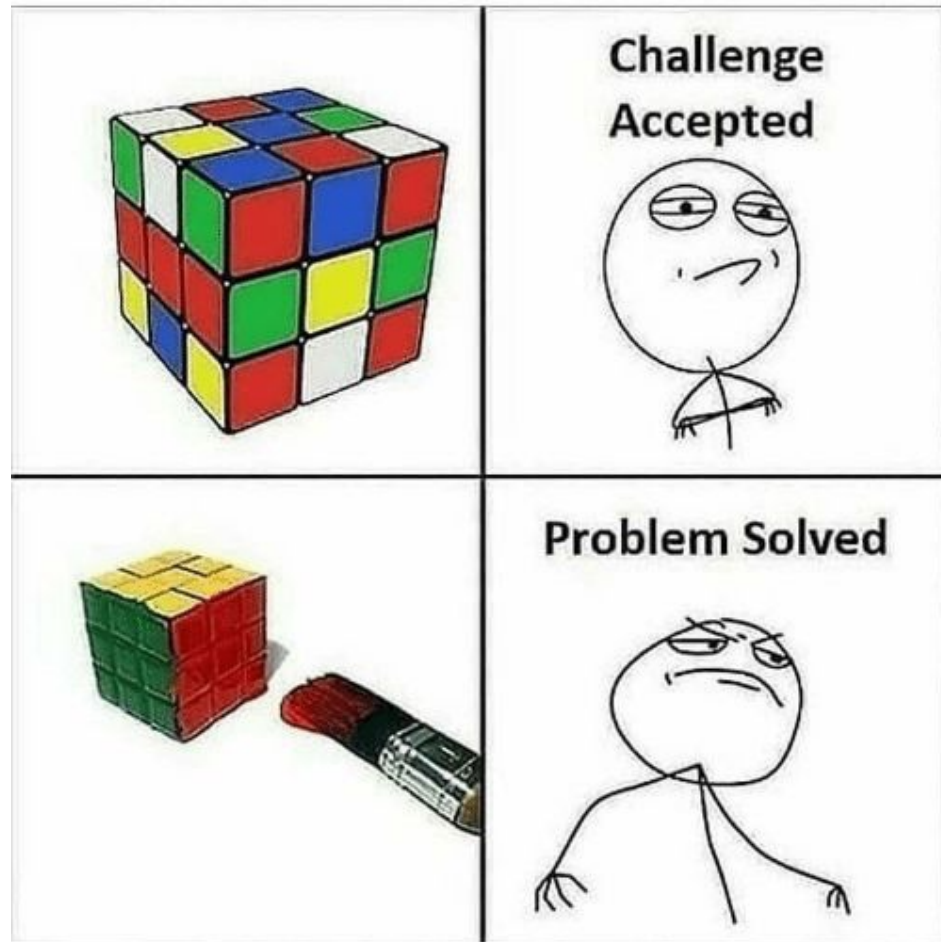
- Com estes conhecimentos de tipos de dados se introduz um formato de troca de dados, similar ao XML.
- Entretanto, é muito mais leve e legível. Vem da sigla **J**ava**S**cript **O**bject **N**otation.
- É basicamente a sintaxe de declaração de objetos JavaScript, porém com a obrigatoriedade de uso de aspas duplas em strings e sem acesso a variáveis globais da linguagem.



JSON

```
{
  "dados": [
    {
      "id": 1,
      "name": "Leanne Graham",
      "username": "Bret",
      "email": "Sincere@april.biz",
      "phone": "1-770-736-8031 x56442",
      "website": "hildegard.org",
      "company": {
        "name": "Romaguera-Crona",
        "catchPhrase": "Multi-layered client-server neural-net",
        "bs": "harness real-time e-markets"
      }
    },
    {
      "id": 2,
      "name": "Ervin Howell",
      "username": "Antonette",
      "email": "Shanna@melissa.tv",
      "phone": "010-692-6593 x09125",
      "website": "anastasia.net",
      "company": {
        "name": "Deckow-Crist",
        "catchPhrase": "Proactive didactic contingency",
        "bs": "synergize scalable supply-chains"
      }
    }
  ]
}
```

CHALLENGE!!



Oh, instagram.....



CHALLENGE!!

- <https://jsbin.com/seyagavole/edit?html,css,js,output>
- <https://github.com/public-apis/public-apis#animals>
- De preferência, sem CORS e autenticação, para facilitar nossa vida :)

