Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

Tecnologias Web
Tabelas

HTML – CSS – JavaScript – PHP - AJAX

João Vieira 2020 – 2021

João Nuno Azevedo, revisão 2023-2024

ipvc estg

CSS.css

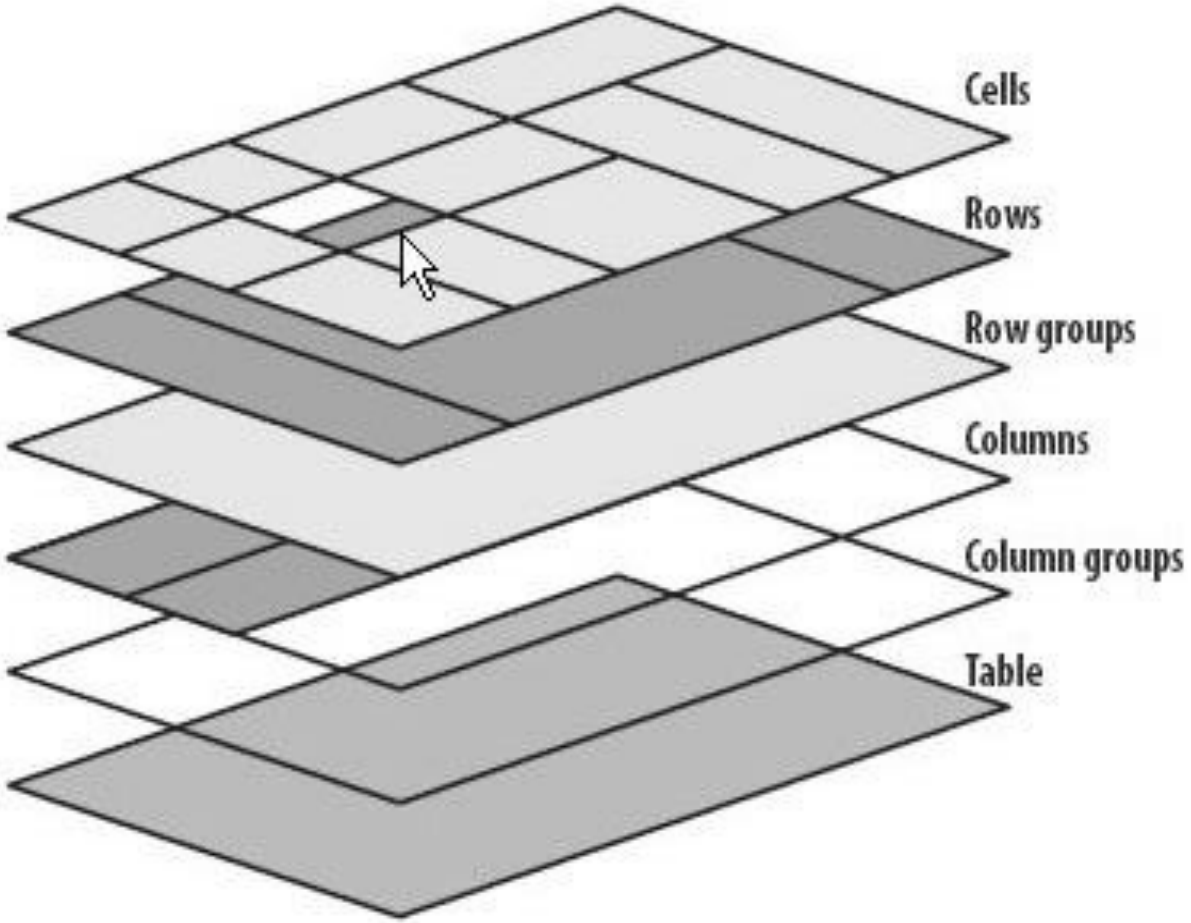
Tabelas - elementos

table	tabela
caption	legenda da tabela
colgroup	grupo de colunas
col	coluna da tabela
thead	cabeçalho da tabela
tfoot	rodapé da tabela
tbody	corpo da tabela
tr	linha da tabela
td	célula da linha da tabela
th	célula da linha do cabeçalho

ipvc estg

CSS.css

Tabelas - camadas de formação



Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

Tabelas - Base

```
<table>
  <colgroup>
    <col style="background-color:lightblue">
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
  <tr>
    <td>5869207</td>
    <td>My first CSS</td>
    <td>$49</td>
  </tr>
</table>
```

ISBN	Title	Price
3476896	My first HTML	$53
5869207	My first CSS	$49

Source:
https://www.w3schools.com/html/html_tables.asp

ipvc estg

CSS.css

Tabelas - Estrutura como exemplo

```
<table>
    <thead>
        <tr>
            <th>Items</th>
            <th>Expenditlure</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>Donuts</th>
            <td>3,000</td>
        </tr>
        <tr>
            <th>Stationery</th>
            <td>18,000</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <th>Totals</th>
            <td>21,000</td>
        </tr>
    </tfoot>
</table>
```

Items	Expenditlure
Donuts	3,000
Stationery	18,000
Totals	21,000

Source:
https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/tfoot

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

Tabelas - colspan e rowspan

```html
<table>
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th colspan="2">Phone</th>
  </tr>
  <tr>
    <td>John Doe</td>
    <td>john.doe@example.com</td>
    <td>123-45-678</td>
    <td>212-00-546</td>
  </tr>
</table>

<h3>Cell that spans two rows:</h3>
<table>
  <tr>
    <th>Name:</th>
    <td>John Doe</td>
  </tr>
  <tr>
    <th>Email:</th>
    <td>john.doe@example.com</td>
  </tr>
  <tr>
    <th rowspan="2">Phone:</th>
    <td>123-45-678</td>
  </tr>
  <tr>
    <td>212-00-546</td>
  </tr>
</table>
```

Cell that spans two columns:

Name	Email	Phone	
John Doe	john.doe@example.com	123-45-678	212-00-546

Cell that spans two rows:

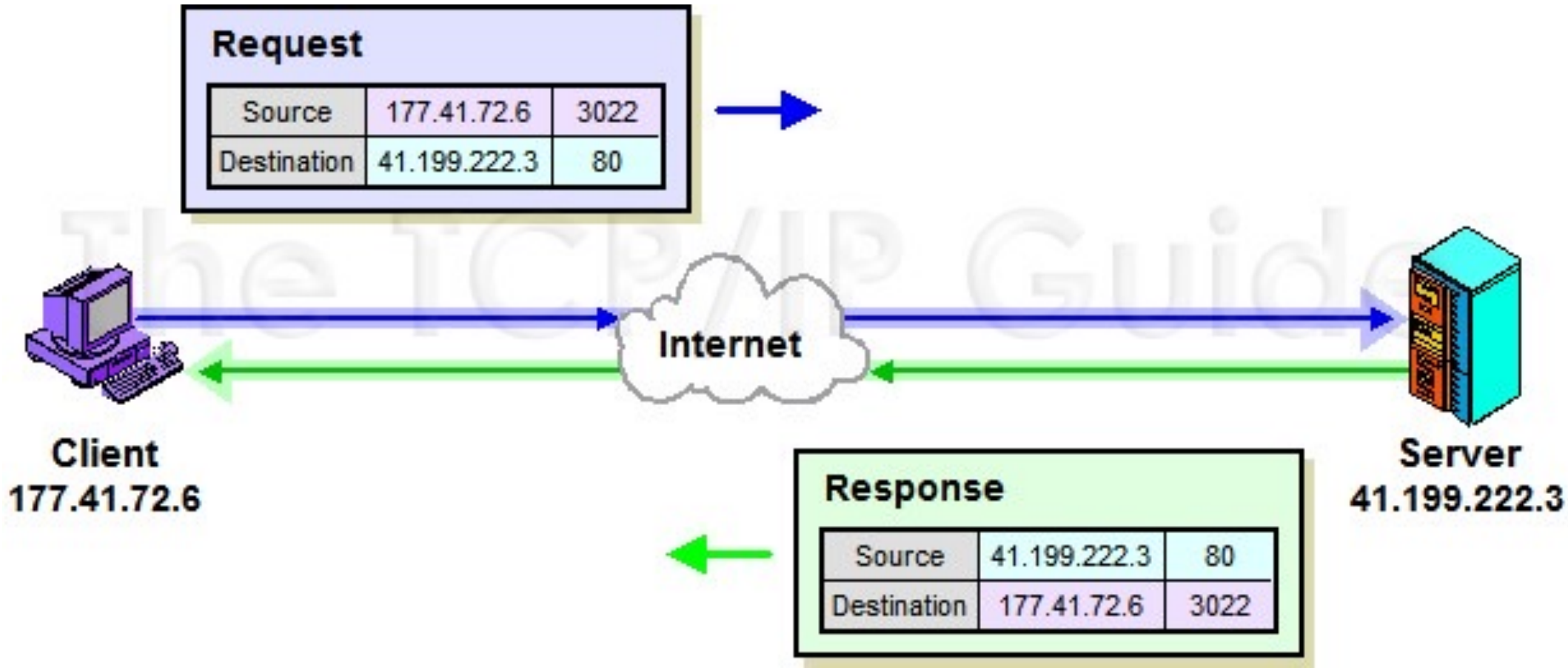
Name:	John Doe
Email:	john.doe@example.com
Phone:	123-45-678
	212-00-546

Source:
https://www.w3schools.com/html/html_table_colspan_rowspan.asp

ipvc estg

CSS.css

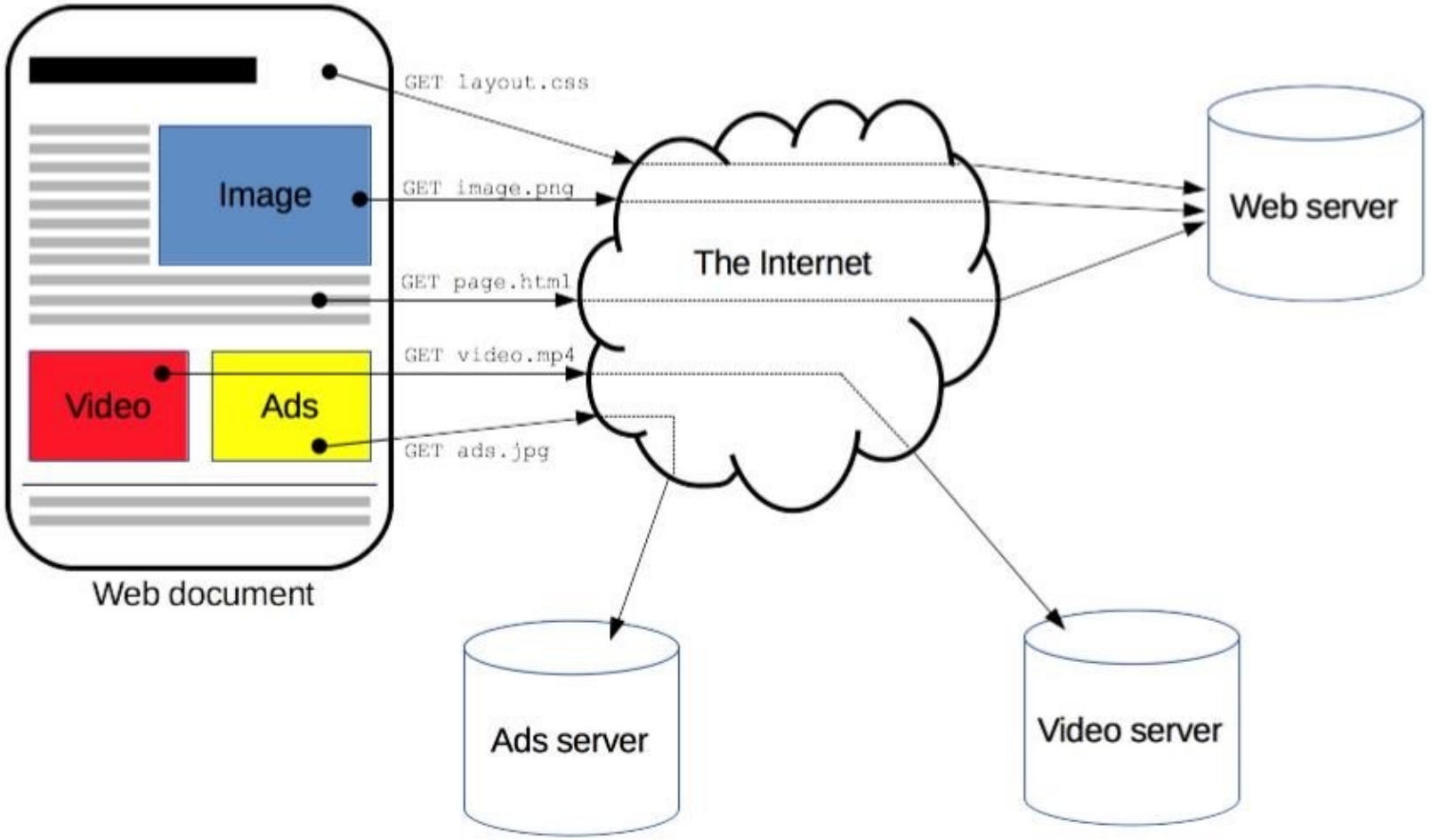
HTTP - Arquitetura cliente-servidor



ipvc estg

CSS.css

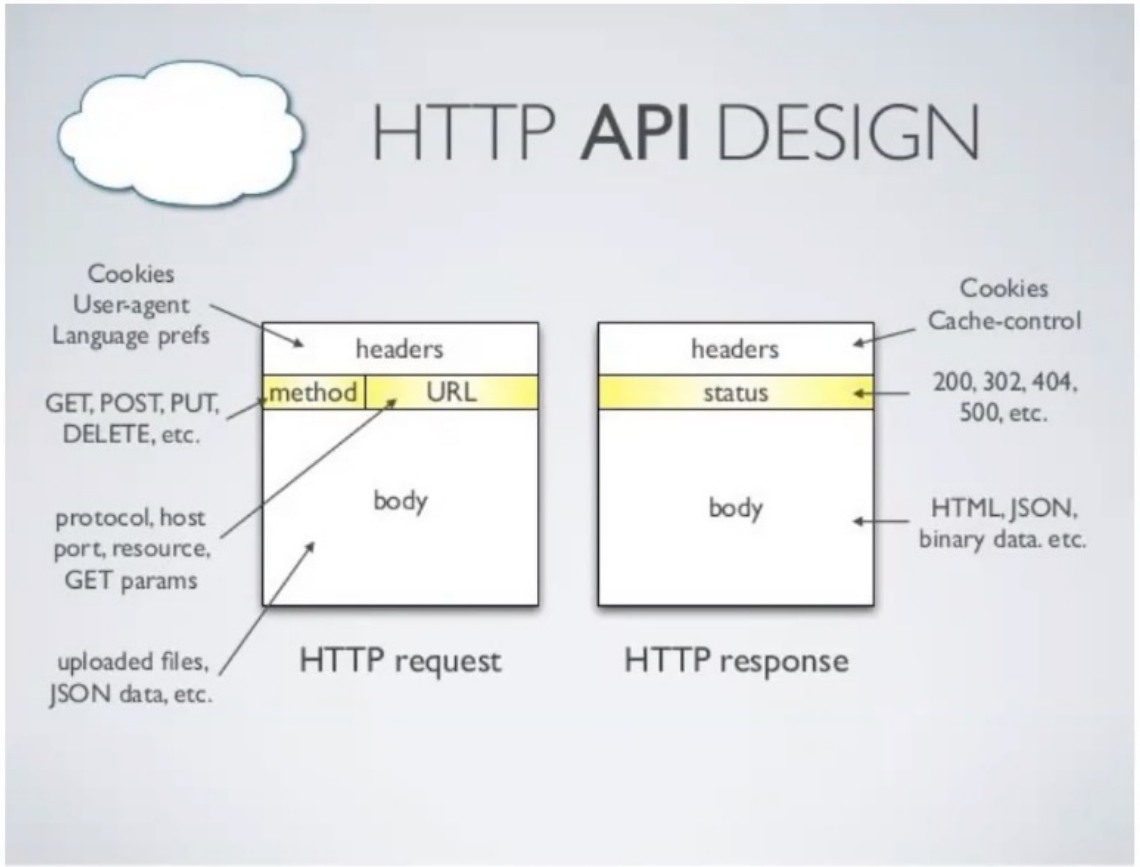
HTTP - Arquitetura cliente-servidor



Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

HTTP - Pedidos e Respostas



Source:
https://gorest.co.in/public/v2/users
https://jsonplaceholder.typicode.com/guide/

ipvc estg

CSS.css

HTTP - Verbos - Ajax

GET

POST

PUT

PATCH

DELETE

etc

https://www.w3schools.com/xml/tryit.asp?filename=tryajax_get

https://www.w3schools.com/xml/tryit.asp?filename=tryajax_get_unique

https://www.w3schools.com/xml/tryit.asp?filename=tryajax_get2

https://www.w3schools.com/xml/tryit.asp?filename=tryajax_post2

Sources:
https://restfulapi.net/http-methods/
https://www.baeldung.com/http-put-patch-difference-spring

ipvc estg

HTTP - Códigos de status de respostas HTTP

Os códigos de *status* das respostas HTTP indicam se uma requisição HTTP foi corretamente concluída.

As respostas podem ser agrupadas desta forma:

1. Respostas de informação (100-199)
2. Respostas de sucesso (200-299)
3. Redirecionamentos (300-399)
4. Erros do cliente (400-499)
5. Erros do servidor (500-599).

200 OK
400 Bad Request
401 Unauthorized
403 Forbidden
404 Not found
405 Method not allowed
500 Internal server error

Sources:
https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status

ipvc estg

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

JavaScript.js

Tecnologias Web

Javascript

HTML – CSS – JavaScript – PHP - AJAX

ipvc estg

CSS.css

JavaScript - scope

```
<script>
    // Global scope


    function myFunction() {
        // Local scope
    }
</script>
```

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

JavaScript - Variáveis - var, let e const

VAR

- **Global** e **Local** scoped
- Podem ser alteradas
- Podem ser declaradas novamente
- Hoisting: true

Sources:
https://www.w3schools.com/js/js_hoisting.asp

ipvc estg

CSS.css

JavaScript - VAR exemplos

```
var global = "teste1";

function print() {
    var local = "teste2";
}

console.log(local); // Erro: "local" is not defined
```

```
var teste = "teste1";
var teste = "teste2";

console.log(teste); // teste2
```

```
console.log(teste); // teste is undefined
teste = "teste";

// é interpretado como

var teste;
console.log(teste); // teste is undefined
teste = "teste";
```

```
var hello = "ola mundo";
var conta = 4;

if (conta > 3) {
    var hello = "hello world";
}

console.log(hello) // hello world
```

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

JavaScript - LET

- **Block** scoped
- Podem ser alteradas
- **Não podem** ser declaradas novamente
- Hoisting: false

ipvc estg

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

JavaScript - LET exemplos

```
let hello = "ola mundo";
let conta = 4;

if (conta > 3) {
    let hello = "hello world";
    console.log(hello) // hello world
}

console.log(hello) // ola mundo
```

```
let teste = "teste1";
let teste = "teste2";

console.log(teste); // Erro: Identifier 'teste' has already been declared
```

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

JavaScript - CONST

- **Block** scoped {}
- **Não podem** ser alteradas, mas as propriedades podem
- **Não podem** ser declaradas novamente
- Hoisting: false

ipvc estg

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

JavaScript - CONST exemplos

```
const hello = "ola mundo";
hello = "hello world";//Erro : Assignment to constant variable.
```

```
const hello = "ola mundo";
const hello = "hello world";//Erro : Identifier 'hello' has already been declared
```

```
const hello = {
    ola: "ola",
    mundo: "mundo"
}

hello = {
    ola: "hello",
    mundo: "world"
} //Erro :  Assignment to constant variable.
```

```
const hello = {
    ola: "ola",
    mundo: "mundo"
}

hello.mundo = "world";
```

ipvc estg

CSS.css

JavaScript - functions

- Regular functions
- Arrow functions

```
hello = function() {
  return "Hello World!";
}
```

- Diferença no THIS

```
hello = () => {
  return "Hello World!";
}
```

- Regular - objeto que invoca a função
- Arrow - proprietário da função

```
hello = () => "Hello World!";
```

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

JavaScript - functions

```
<body>
    <button id="btn">Isto é um botão</button>
</body>

<script>
    var hello = "hello world"

    function regular() {
        console.log(this.hello) // unde
        console.log(this.innerText)
    }

    arrow = () => {
        console.log(t
    }

    docume
    do
</
```

ipvc estg

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

JavaScript - Mutações

- Arrays
- Objects

```
const person = {
    name: 'John Doe',
    email: 'john@doe.com'
};
const samePerson = person;

person.name = "Jane Doe";

console.log(samePerson); // {name: "Jane Doe", email: "john@doe.com"}
```

```
const cities = ['Oslo', 'Rome', 'Cork', 'Paris', 'London', 'Bern'];
const copy = cities;

cities.splice(2);

console.log(copy); // ['Oslo', 'Rome']
```

ipvc estg

CSS.css

JavaScript - Mutações

- Push ✗
- Unshift ✗
- Splice ✗
- Pop ✗
- Fill ✗
- Reverse ✗
- Concat ✓
- Slice ✓
- Es6 spread operator ✓

```
const numbers = [1, 2];
const moreNumbers = [...numbers, 3];
```

Sources:
https://www.w3schools.com/jsref/jsref_obj_array.asp

ipvc estg

CSS.css

JavaScript - Mutações

- Direct addition ✗
- Object assign ✓
- Es6 spread operator ✓

```
const person = {name: 'John Doe', email: 'john@doe.com'};
const samePerson = {...person};
const copy = Object.assign({}, person);

console.log(person); // { name: 'John Doe', email: 'john@doe.com' }
console.log(samePerson); // { name: 'John Doe', email: 'john@doe.com' }
console.log(copy); // { name: 'John Doe', email: 'john@doe.com' }
```

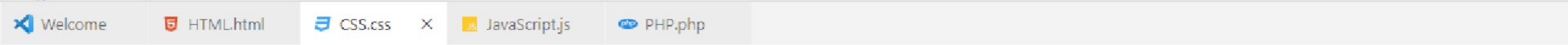
CSS.css

JavaScript - Arrays

usefull Functions

- **map** - Creates a new array with the result of calling a function for each array element
- **forEach** -Calls a function for each array element
- **filter** - Creates a new array with every element in an array that pass a test
- **concat** - Joins two or more arrays, and returns a copy of the joined arrays
- **some** - Returns true or false if passes the test
- **find** - Returns the value of the first element in an array that pass a test
- **findIndex** - Returns the index of the first element in an array that pass a test
- **indexOf** - Search the array for an element and returns its position

ipvc estg



CSS.css

JavaScript – Some Best Practices

- Minimize the use of global variables.
- All variables used in a function should be declared as **local** variables.
- Initialize Variables
- Use === Comparison
- Assign default values to arguments

- Reduce Activity in Loops
- Reduce DOM access
- Avoid duplicated or unnecessary variables

ipvc estg

Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

JavaScript - Storage (local & session)

Guardar Informação no browser, mas apenas na sessão, isto é se trocar de tab por exemplo esta storage é perdida

```
// Store
sessionStorage.lastName = "Smith";

// Retrieve
sessionStorage.lastName;

// Delete
sessionStorage.removeItem("lastname");
```

Guardar Informação no browser, mas apenas no browser, logo é possivel utilizar a mesma storage em várias tabs.

```
// Store
localStorage.setItem("lastname", "Smith");

// Retrieve
localStorage.getItem("lastname");

// Delete
localStorage.removeItem("lastname");
```

ipvc estg

CSS.css

JavaScript - Storage (local & session e Cookies)

	Cookies	Local storage	Session storage
Capacity	4KB	10MB	5MB
Accessible from	Any window	Any window	Same tab
Expiration	Manually set	Never	On tab close
Storage location	Browser and server	Browser only	Browser only
Sent with requests	Yes	No	No
Blockable by users	Yes	Yes	Yes
Editable by users	Yes	Yes	Yes

Source:
https://www.30secondsofcode.org/articles/s/cookies-local-storage-session