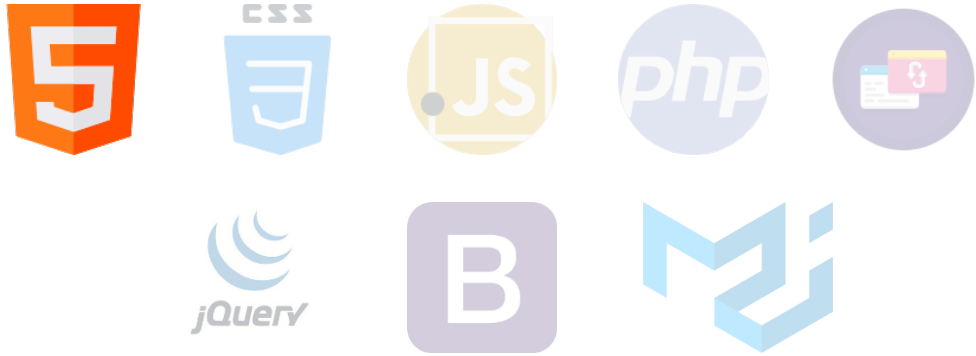
Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

HTML.html

Tecnologias Web

HTML

HTML – CSS – JavaScript – PHP - AJAX



João Vieira 2020–2021

João Azevedo, revisão 2023-2024

ipvc estg

HTML.html

HTML - Definições

- ***Documento HTML (HyperText Markup Language Document)***
  - um ***documento HTML*** é um documento SGML que está de acordo com a especificação HTML.

- ***Autor (Author)***
  - Um ***autor*** é uma pessoa ou programa que escreve ou gera ***documentos HTML***.
  - Uma ***ferramenta de autoria*** é um caso especial de um ***autor***, mais precisamente, é um programa que gera HTML.

HTML.html

HTML - Definições

- ***Utilizador (User)***
  - um ***utilizador*** é uma pessoa que interage através de um ***agente*** para ver, ouvir ou usar de uma qualquer outra forma um ***documento HTML*** interpretado (“*rendered*”).

- ***Agente HTML (HTML user-agent)***
  - Um ***agente HTML*** é um qualquer dispositivo que interpreta ***documentos HTML***. Os “browsers” visuais, os “browsers” não visuais (audio, braille), os proxies, robots de pesquisa, etc, são ***agentes HTML***.

HTML.html

HTML - Definições

- ***Descontinuado (Deprecated)***
  - Elementos ou atributos **descontinuados** são aqueles, que fazendo parte de especificações ou versões anteriores, foram substituídos por novos elementos, atributos ou construções.
  - Os elementos e atributos **descontinuados** continuam a ser suportados, de forma transitória, pelos **agentes HTML**.
- ***Obsoleto (Obsolete)***
  - um elemento ou atributo ***obsoleto*** é aquele para o qual já não existem garantias de suporte por parte de um ***agente HTML***.

HTML.html

HTML - Definições

- ***Elementos e Marcadores (Elements and Tags)***
  - ***Elementos*** são as estruturas que descrevem partes de ***documentos HTML***.
    - EX:
      - Elemento **P** representa um parágrafo
      - Enquanto o elemento **EM** dá ênfase ao seu conteúdo.
  - Um elemento possui três partes:
    - um marcador de início (openning tag);
    - conteúdo (content);
    - marcador de fim (closing tag) (alguns elementos podem não ter closing tag).

HTML.html

HTML - Definições

- Um ***marcador*** (***tag***) é um texto (especial) delimitado por "<“ e ">".
- Um **marcador de fim** inclui uma "/" depois do "<".
  - Exemplo:
    - o elemento EM possui um marcador de início <EM>, e um marcador de fim </EM>.
    - Os marcadores de início e fim delimitam o conteúdo do elemento EM:

<EM> texto com ênfase </EM>

- Os nomes dos elementos não são sensíveis a maiúsculas (**case-insensitive**) então: , <eM>, and <EM> são todos representações adequadas do elemento.
- **Os elementos não se devem misturar**. Se o marcador de início para um elemento EM ocorre dentro do conteúdo de um elemento P, então o marcador de fim desse elemento EM deve ocorrer dentro do conteúdo do mesmo elemento P.

ipvc estg

HTML.html

HTML - Definições

- Alguns elementos permitem que os marcadores de início e/ ou de fim sejam omitidos.
  - Exemplo, o marcador de fim do elemento P é opcional, dado que o fim deste elemento é implicitamente determinado pelo início do próximo elemento P:

    <P> um parágrafo

    <P> outro parágrafo

- **Alguns elementos não possuem marcador de fim** pois não possuem conteúdo.
  Estes elementos, tal como o BR, ou elementos de **forms**, são representados apenas pelo marcador de início e dizem-se elementos vazios **(empty elements).**

HTML.html

HTML - Definições

- ***atributos (attributes)*** *ou* ***props (properties)***
  - Os **atributos de elementos definem diversas propriedades para esse elemento**.
  - Exemplo:

    o elemento IMG possui um atributo SRC para fornecer a localização do ficheiro de imagem e um atributo ALT para uma descrição alternativa em texto caso a imagem não seja visualizada:

    <img **src**=”estglogo.gif**" alt=”**Logotipo da ESTG**">**

- Um atributo está incluído sempre no marcador de início e toma a forma:

  nome-atributo =”valor-do-atributo“

- O valor do atributo é delimitado por aspas. Estas podem ser opcionais em determinados casos.
- Os nomes dos atributos são insensíveis a maiúsculas (case-insensitive), mas os valores podem ser sensíveis (case-sensitive).

ipvc estg

HTML.html

HTML - Definições

- **universal resource identifier (URI) – identificador de recursos**
  - cada recurso disponibilizado na Internet (documentos HTML, imagens, vídeos, programa, etc), possui um endereço.
  - este endereço pode ser codificado numa URI (universal resource identifier)
  - uma URI tem tipicamente três partes:
    - o **mecanismo de acesso ao recurso** (tipicamente o protocolo de acesso: http, ftp, etc..);
    - a identificação (endereço lógico ou **endereço físico**) do servidor que aloja o recurso;
    - o path (caminho) até ao recurso;

HTML.html

HTML - Definições

- **URI exemplos (fonte wikipedia)**
  - **ftp**://ftp.is.co.za/rfc/rfc1808.txt
  - **gopher**://spinaltap.micro.umn.edu/00/Weather/California/Los%20Angeles
  - **http**://www.linux.ime.usp.br/~andrew/mac499/index.htm
  - **mailto**:andrew@linux.ime.usp.br
  - **news**:comp.infosystems.www.servers.unix
  - **telnet**://melvyl.ucop.edu/

HTML.html

HTML - Atributos / Props

Um determinado número de atributos em HTML 4.0 são comuns à maioria dos elementos. Estes atributos comuns dividem-se em:

- **Core props**
  - ID
  - CLASS
  - STYLE
  - TITLE
  - etc…
- **Internationalization**
  - LANG
  - DIR
  - etc…
- **Events / Scripting**
  - ONCLICK
  - ONMOUSEOVER
  - ONMOUSEOUT
  - etc..

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

HTML.html

Tecnologias Web

CSS

HTML – CSS – JavaScript – PHP - AJAX

ipvc estg

CSS.css

CSS - Cascading Style Sheet

Uma folha de estilos ("**style sheet**") é constituída por regras que indicam aos agentes ("browsers") a toda a parte visual, de cor e formatações, na apresentação de um documento.

- **Modos de Utilização:**
  - Inline;
  - Class;
  - Id;
  - Elementos;
  - Selectores.

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

CSS.css

CSS - Cascading Style Sheet

- CSS - Hierarquia de efeito cascata
  - Declarações com !Important
  - Aplicações de estilos
    - Inline
    - Incorporado <style> </style> dentro do <head>
    - Externo (importado localmente ou por link)
    - Folha de estilo do utilizador
    - Folha de estilo do navegador (uma extensão do browser pode ter regras definidas)
  - Herança
    - Inline, id, class, selectores e atributos

ipvc estg

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

CSS.css

CSS - Cascading Style Sheet

- CSS -Seletores
  - div p (todos os elementos)
  - div > p (filhos diretos)
  - div ~ p (todos após)
  - div + p (adjacência)
  - * (todos)
  - #myid (elemento com id único)
  - .myclass

ipvc estg

CSS.css

CSS - Cascading Style Sheet

Pseudo classes

::after	::before	::first-letter
::first-line	::selection	::backdrop
::placeholder	::marker	::spelling-error
::grammar-error		

SOME MORE PSEUDO-CLASSES

:active	:link	:not()
:checked	:disabled	:focus
:any	:checked	:default
:disabled	:empty	:enabled
:first	:first-child	:first-of-type
:focus	:hover	:last-of-type
:not()	:nth-child()	:nth-last-child()
:nth-last-of-type()	:nth-of-type()	:only-child
:required	:right	:root

ipvc estg

CSS - Cascading Style Sheet

- Length – unidades de medida:

- Absoluta
  - in (polegadas, 1in = 2.54cm)
  - mm (milímetros)
  - cm (centímetros)
  - pt (points, 1pt = 1/72in)
  - pc (picas, 1pc = 12pt)

- Relativa
  - px (pixels, depende da resolução do monitor)
  - em (ems, a altura da fonte utilizada no elemento parent)
  - % (percentagem relativa aos elementos envolventes)

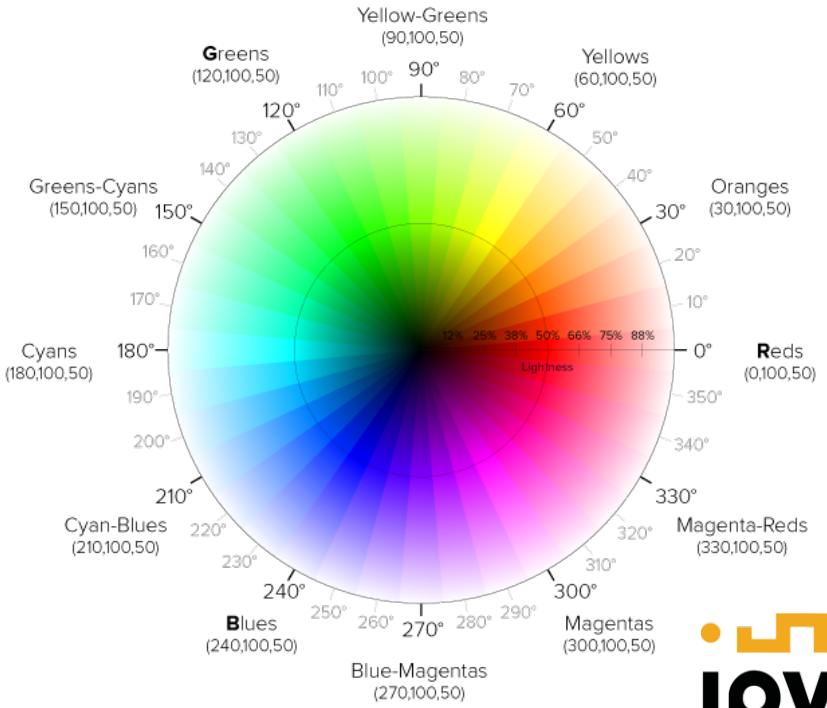
ipvc estg

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

CSS.css

CSS - Cascading Style Sheet - **Cor**

- Existem 3 maneiras fundamentais de definir uma cor em css:
  - **rgb(red, green, blue)**
    - ex: rgb(255, 0, 0) (this will be red);
    - ex: rgb(0, 255, 0) (this will be green);
    - ex: rgb(0, 0, 255) (this will be blue).
  - **#rrggbb** (red, green, blue)
    - ex: #FF0000 (this will be red);
    - ex: #00FF00 (this will be green);
    - ex: #0000FF (this will be blue).
  - **hsl(hue, saturation, lightness)**
    - ex: hsl(0, 100%, 50%) (this will be red);
    - ex: hsl(120, 100%, 50%) (this will be green);
    - ex: hsl(240, 100%, 50%) (this will be blue).



ipvc estg

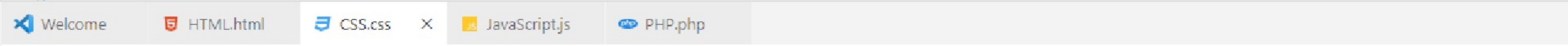
Welcome | HTML.html | CSS.css | JavaScript.js | PHP.php

CSS.css

CSS - Cascading Style Sheet - **URL**

- Exemplos:
  - body { background: **url(stripe.gif)** }
  - body { background: **url(http://www.htmlhelp.com/stripe.gif)** }

ipvc estg



CSS.css

CSS - Cascading Style Sheet - **Fontes**

- Exemplos:
  - font-family (fonts a utilizar)
  - font-style (italic, oblique)
  - font-weight (grossura, tipo bold, narrow ou mesmo valores 100 ou 900)
  - font-size (em qualquer unidade de medida)

ipvc estg

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

CSS.css

CSS - Cascading Style Sheet - **Cor e Fundo**

- Exemplos:
  - color (pode ser aplicado a texto e icons .svg)
  - background-color
  - background-image
  - background-repeat
  - background-position
  - background

ipvc estg

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

CSS.css

CSS - Cascading Style Sheet - **Texto**

- Espaçamentos de texto
- Exemplos:
  - text-align
  - word-spacing
  - letter-spacing
  - text-decoration
  - text-transform
  - text-indent
  - line-height

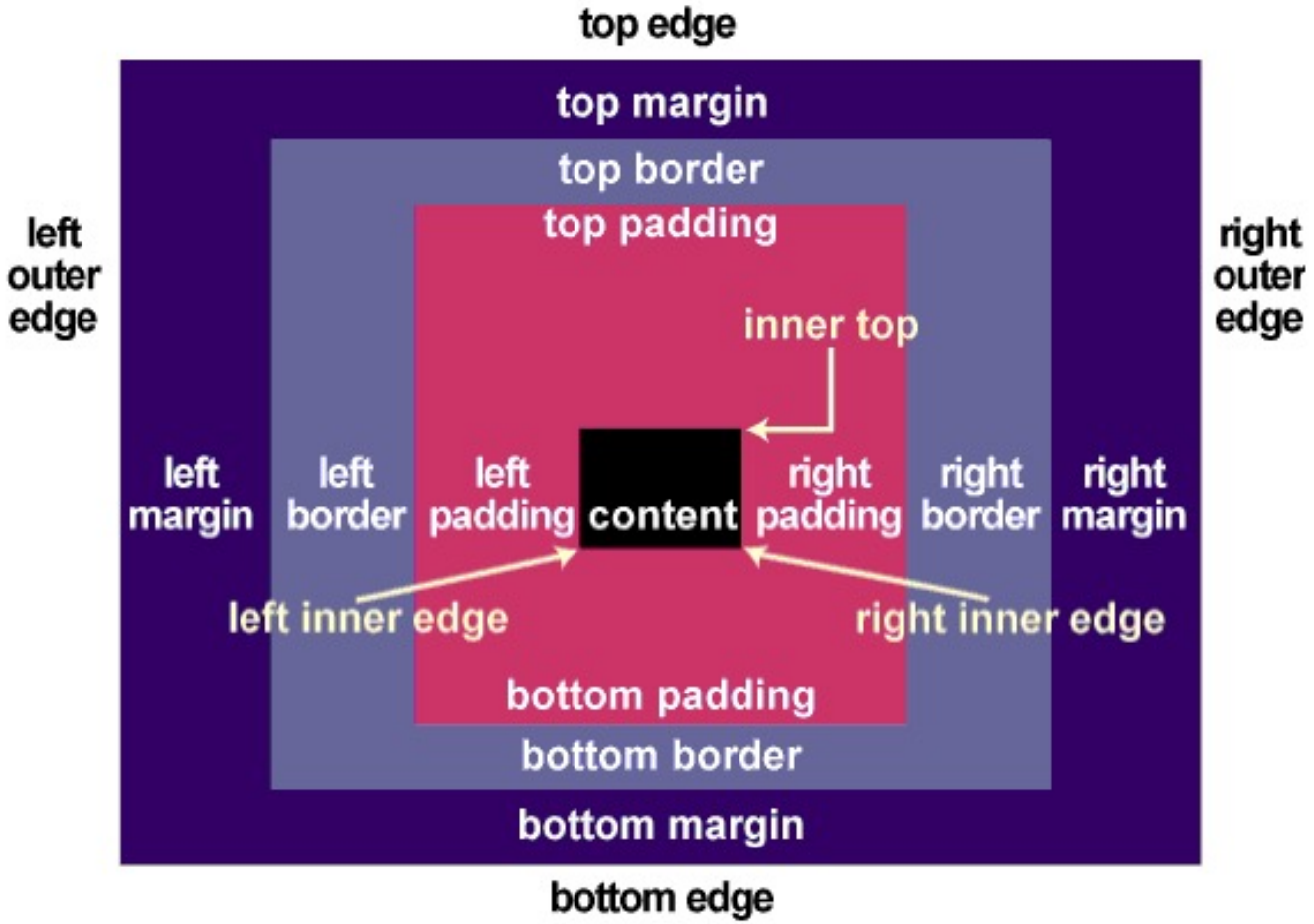
ipvc estg

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

CSS.css

CSS - Cascading Style Sheet - **Caixas**

- Exemplos:
  - Margin
  - Border
  - Padding



ipvc estg

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

HTML.html

Tecnologias Web

DOM

HTML – CSS – JavaScript – PHP - AJAX

ipvc estg

HTML.html

DOM - Document Object Model

- É uma **interface de programação para páginas HTML e XML**;
- Onde temos **acesso à estrutura do "documento"** da página e, assim **podemos realizar ações** como adicionar, remover ou alterar **elementos** do seu conteúdo;
- Temos acesso tanto ao conteúdo da página em si, assim como o que envolve esse conteúdo, como por exemplo algumas configurações, headers, url's e funções;
- O conteúdo da página que pode ser modificado, vai desde os elementos em si, como os seus atributos, os seus estilos, os seus eventos, valores etc.

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

HTML.html

DOM - Document Object Model

- **Obter elementos:**
  - document.getElementById(id)
  - document.getElementByTagName(tagName)
  - document.getElementByClassName(className)

- **Alterar elementos:**
  - element.innerHTML = New Payload (any type)
  - element.attribute = new value
  - element.style.property = new style
  - element.setAttribute(attibute, value)

ipvc estg

HTML.html

DOM - Document Object Model

- **Adicionar/Remover elementos**
  - document.createElement(element)
  - document.removeChild(element)
  - document.appendChild(element)
  - document.replaceChild(newElement, oldElement)

- **Eventos:**
  - onclick
  - onload
  - onunload
  - onmouseout
  - onmousedown
  - onmouseup
  - etc...

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

HTML.html

Tecnologias Web
BOM

HTML – CSS – JavaScript – PHP - AJAX

ipvc estg

HTML.html

BOM - Browser Object Model

- É uma **camada acima da DOM** onde temos acesso a outras propriedades abstratas ao documento/website.

- Exemplo:
  - **screen.width**

HTML.html

BOM - Browser Object Model

- Funcionalidades do BOM:
  - **window** - representa a janela do browser;
  - **screen** – contém informação sobre o ecrã do utilizador;
  - **location** – navegação entre páginas;
  - **history** – histórico do browser relativo à página em que estamos;
  - **navigator** – representa o estado e identidade do user-agente;
  - **timming** – funções relacionadas com tempo;
  - **popup alert** – janelas de alerta, confirmação e inserção;
  - **cookies** – gerir cookies do utilizador.
    - document.cookie

HTML.html

BOM - Browser Object Model - **Window**

Funções do Window

- window.innerHeight(body)
- window.innerWidth(body)
- window.open()
- window.close()
- window.moveTo()
- window.resizeTo()
- Outras: https://www.w3schools.com/jsref/met_win_open.asp

HTML.html

BOM - Browser Object Model - **Screen**

- window.screen.width || screen.width
- screen.height
- screen.availWidth
- screen.availHeight
- screen.colorDepth
- screen.pixelDepth

Exemplos e referências:

https://www.w3schools.com/jsref/prop_win_screen.asp

HTML.html

BOM - Browser Object Model - **Location**

- window.location.href || location.href
- window.location.href
- window.location.href
- window.location.protocol

Exemplos e referências:

https://www.w3schools.com/jsref/prop_win_location.asp

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

HTML.html

BOM - Browser Object Model - **History**

- window.history || history.back()
- history.forward()

Exemplos e referências:

https://www.w3schools.com/jsref/prop_win_history.asp

ipvc estg

HTML.html

BOM - Browser Object Model - Timming

- window.setTimeout(*function*,*milliseconds*);
- window.setInterval(*function*,*milliseconds*);
- window.clearTimeout(*timeoutVariable*)
- window.clearInterval(*timerVariable*)
- New Date()

Exemplos e referências:

https://www.w3schools.com/jsref/met_win_setinterval.asp

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

HTML.html

BOM - Browser Object Model - **Popup alert**

- window.alert()
- window.confirm()
- window.prompt()

Exemplos e referências:

https://www.w3schools.com/jsref/met_win_alert.asp

ipvc estg

HTML.html

BOM - Browser Object Model - **Cookies**

- document.cookie="username=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";
- decodeURIComponent(document.cookie)

Exemplos e referências:

https://www.w3schools.com/js/js_cookies.asp
https://www.w3schools.com/jsref/jsref_decodeuricomponent.asp

ipvc estg

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

HTML.html

Tecnologias Web

JSON e XML

HTML – CSS – JavaScript – PHP - AJAX

ipvc estg

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

HTML.html

JSON - JavaScript Object Notation

- Utilização:

```
myObj = {
  "name":"John",
  "age":30,
  "cars": {
    "car1":"Ford",
    "car2":"BMW",
    "car3":"Fiat"
  }
 }
utilizador.nome

utilizador['nome']
```

- Um objeto **JSON é construído sempre com "chave":"valor"**;
- Pode iniciar com {} ou [] dependendo dos dados
- Serve para guardar e usar informação de uma forma limpa, fácil e eficiente;
- Enviar e receber dados em comunicação com servidores ou como fonte de dados.

ipvc estg

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

HTML.html

JSON - JavaScript Object Notation

- Em **Json**, podemos ter **qualquer tipo de valores**:
  - uma String
  - uma number
  - outro object (JSON object)
  - um array
  - valor boolean
  - Null

ipvc estg

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

HTML.html

XML - eXtensible Markup Language

- Utilização:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

- Serve para guardarmos e usar informação;
- Enviar e receber dados em trocas com servidores
- Complemento ao HTML

ipvc estg

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

HTML.html

JSON vs XML

- Semelhanças:
  - Fácil interpretação
  - Hierarquia
  - Múltiplas linguagens de programação
  - Transporte de dados

ipvc estg

HTML.html

JSON vs XML

- Diferenças:
  - JSON não usa tags, por isso não precisa de as fechar
  - **JSON é mais simples**
  - **JSON é mais fácil de ler e escrever**
  - JSON é mais rápido no transporte (JSON = String; XML = document)
  - **JSON é mais fácil fazer parse**

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

JQUERY

- Utilização:
  - Manipulação HTML/DOM
  - Manipulação CSS
  - Eventos
  - Efeitos e animações
  - Pedidos Ajax

```
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
```

ipvc estg