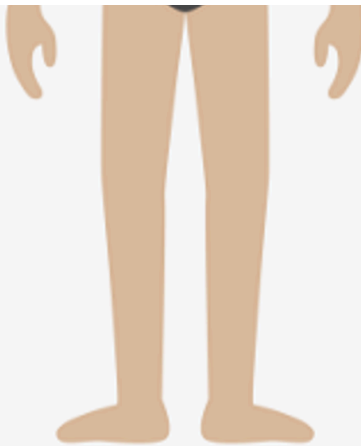
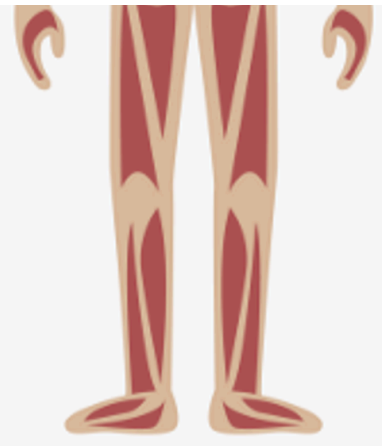


HTML



CSS



Javascript

Linguagem de Programação II

Link para aulas

- <https://u.pcloud.link/publink/show?code=kZFajwVZAgJbR8eJaa4FMBBqOpfktBWP27uk#folder=18360176661&tpl=publicfoldergrid>

Primeiro Bimestre

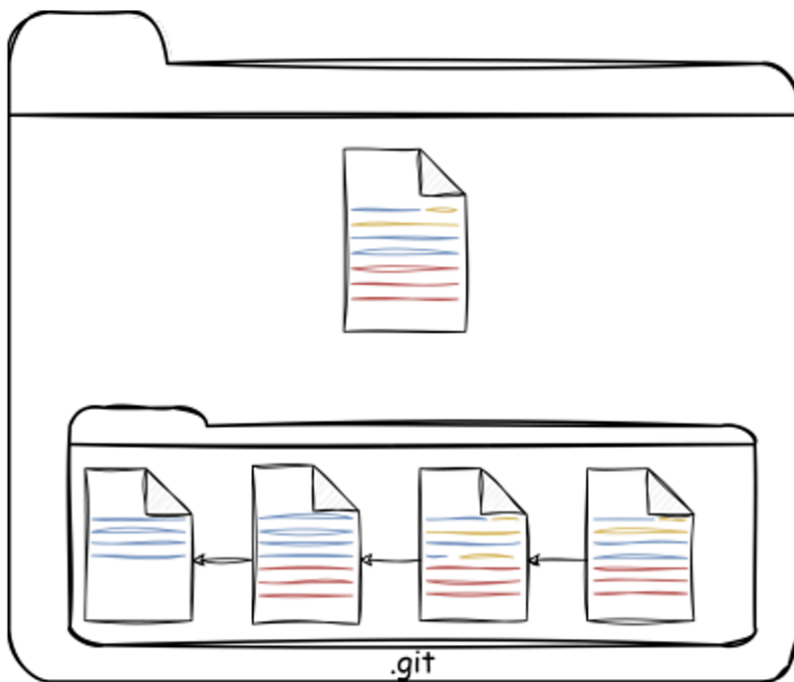
Aula 01 - git

Introdução

- Controle de versão
 - acessar partes removidas no futuro
 - controle de versão manual
 - mesclar conteúdo produzido por pessoas diferentes em uma única versão
 - esquecer de criar uma nova versão do arquivo antes de editar conteúdo existente
 - perder conteúdo ao sobrescrever um arquivo sem intenção
 - criação de sistemas de controle de versão → **git**

Usando o *git*

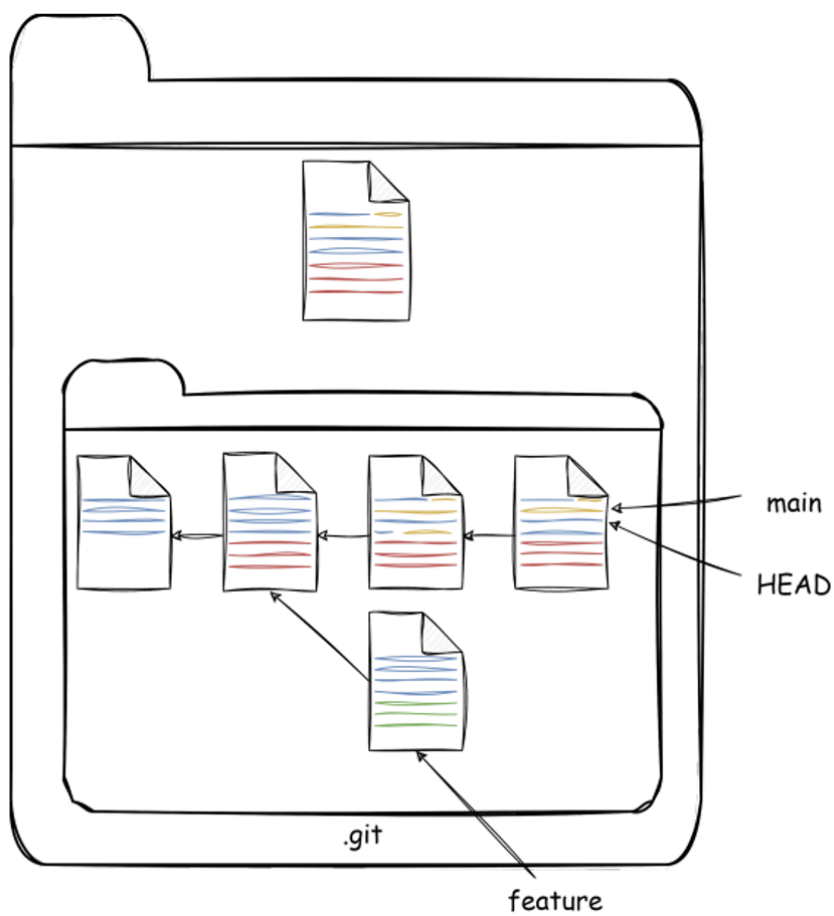
- deve-se criar uma pasta para arquivos de interesse → repositório
- pasta para registrar informações sobre controle de versões → repositório git / .git
- tornar alterações permanentes → operação commit
 - cada commit criará uma versão nova do arquivo editado, que foi commitado



Branches e ponteiro HEAD

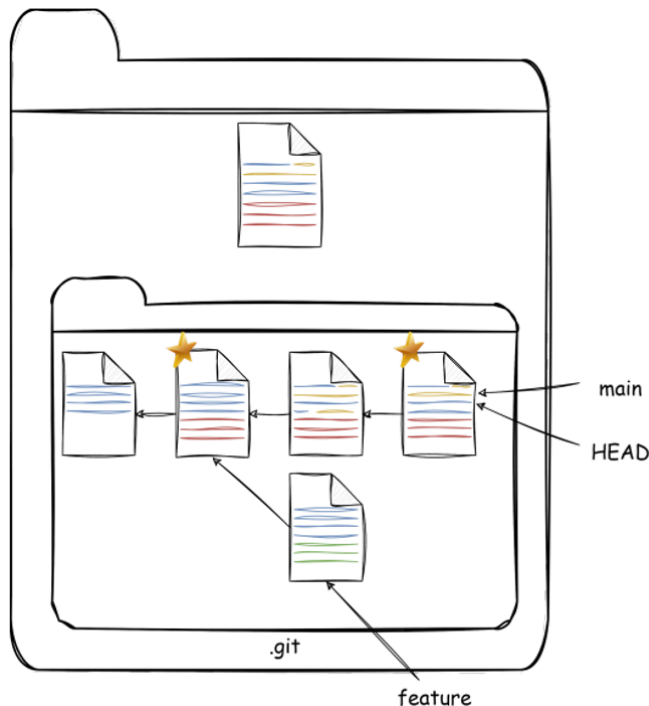
- ponteiro que faz referência à versão atual → HEAD

- ponteiro para um determinado commit → branch
 - gera-se um "rumo" a ser seguido
 - ex: **branch** *main* e *feature*



Tags

- destaque de um commit → tag
 - representa algo notável



- tipos
 - leves: ponteiro temporário para algum commit
 - anotada: destaca um commit, incluindo:
 - nome do autor
 - mensagem
 - data
 - etc...
 - <https://git-scm.com/book/en/v2/Git-Basics-Tagging>

Repositórios Remotos (remotes)

- repositório feito para compartilhar com outras pessoas → Repositórios Remotos
 - trabalho simultâneo
 - backup de todo o controle de versão
 - exemplos
 - *github.com*
 - *gitlab.com*
 - *bitbucket.org*

- upload do conteúdo do repositório local no remoto → push
- variável que armazena o link do repositório remoto → origin
 - vem de "originally"
 - "repositório local foi **originalmente** clonada do repositório remoto"

Comandos iniciais

```
# Início git init # criação da pasta .git # Configurações iniciais git
config --local user.name "EXEMPLO_DE_NOME" # configura nome responsável
pelas # operações feitas git config --local user.email "EXEMPLO_DE_EMAIL"
# configura email responsável pelas # operações feitas git config --global
core.editor "code --wait" # configurando o VS CODE como editor # das
mensagens de commit git config --global init.defaultBranch main #
configurando "main" como nome padrão # para novas branches git config --
list # verifica configurações feitas
```

Estados de arquivos

- Untracked
 - arquivos no diretório sem controle de versão
- Tracked
 - // // // com // // //
- Unmodified
 - versão do diretório é igual à última tornada permanente
- Modified
 - // // // está editada em relação à última tornada permanente
- Staged
 - arquivos prontos para participar do próximo commit

Implementação de uma calculadora (exemplo)

- criação de um arquivo *calculadora.py*

```
git status # saber estado atual do arquivo git status -s # mais curto git
add calculadora.py # torna o arquivo staged git commit # abrirá uma aba no
VS CODE para digitar o commit git commit -m "apostila02(git-python):
funcao somar implementada"
```

- Conventional Commits
 - <https://github.com/angular/angular/blob/main/CONTRIBUTING.md#commit>
- criação do arquivo *teste_calculadora.py*

```
git status -s git add teste_calculadora.py git status -s git commit -m
"apostila02(git-python): teste da funcao somar"
```

- adição de uma função de subtrair em *calculadora.py*

```
git diff # verifica arquivos modificados em relação ao original
(Modified)
```

- tornar a operação de soma (commit anterior) "especial"

```
git tag -a v1.0.0 -m "operação de somar implementada e testada" #
aplica-se uma tag ao commit atual git tag --list # verifica todas as
tags git show v1.0.0 # verifica especificamente a tag "v1.0.0" git
commit -a -m "apostila02(git-python): funcao subtrair implementada" #
committa a subtração
```

- adiciona no *teste_calculadora.py* a função de subtração

```
git add teste_calculadora.py git commit -m "apostila02(git-python):
teste da funcao subtrair" git tag -a v1.0.1 -m "operação de subtrair
implementada e testada" git tag git show v1.0.0 git show v1.0.1
```

Repositórios remotos com *GitLab*

- fazendo o "upload" do repositório local
- geração de token de autenticação
 - não se deve digitar as credenciais para login a cada acesso
 - `https://oauth:token@gitlab.com/usuário/repositório.git/`
 - ex:

https://oauth:glpat-xJ8xh7spaiEDQkr3i5Bj@gitlab.com/professorbossini/20231_pessoal_m_aua_vgti_tutorial_git_python.git/

```
git remote add origin $TOKEN # adiciona o token de autenticação na
variável "origin" git remote remove origin # retira token git remote #
verifica o nome do remote git remote -v # verifica o nome do remote e o
link git branch # verifica nome da brach git push origin main # faz o push
para repositório remoto, SEM AS TAGS git push origin v1.0.0 # faz o push
com uma tag em específico git push origin --tags # faz o push com todas as
tags
```

Aula 02 - HTML

Elementos e tags HTML

- tag
 - abertura, conteúdo, fechamento
 - exibição no navegador varia conforme seu tipo

Listas e alinhamento

```
<h1>Lista de produtos</h1> <ul> <!-- unordered list --> <li>Arroz</li> <!--
- list item --> <li>Feijão</li> <li>Laranja</li> </ul>
```

Sublistas

```
<h1>Lista de produtos</h1> <ul> <li>Arroz</li> <li>Feijão</li> <li> Frutas
<ul> <li>Laranja</li> <li>Pêssego</li> <li>Maçã</li> </ul> </li> </ul>
```

Estrutura de página HTML

- *html*: primeiro elemento
- filhos
 - *head*: abriga metadados
 - define título da página, por ex
 - *body*: abriga conteúdo visível

Encoding

- mapeamento de um conjunto de caracteres para sequência de bits
 - ex: UTF-8
- quando arquivo é salvo → editor de texto usa encoding para armazenar conteúdo
 - sequência de bits
- é possível especificar um metadado que indica qual encoding foi utilizado
 - ele não salva com o encoding especificado, o editor de texto que decide qual encoding utilizar
 - navegadores mais modernos serão capazes de deduzir essa informação a partir do conteúdo inicial dos documentos

```
<html> <head> <title>Documentos HTML</title> <meta charset="utf-8">
</head> <body> <h1>Estrutura de um documento HTML</h1> <p>Estamos
usando os elementos html, head e body.</p> </body> </html>
```

Versões do HTML

- especificando versão 5 do HTML

```
<!DOCTYPE html> <html> <head> <title>Documentos HTML</title> <meta
charset="utf-8"> </head> <body> <h1>Estrutura de um documento HTML</h1>
<p>Estamos usando os elementos html, head e body.</p> </body> </html>
```

- não é *case sensitive*

Acessibilidade: especificando idioma da página

- dispositivos capazes de ler o conteúdo e reproduzir sonoramente para deficientes

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <title>Documentos HTML</title>  
<meta charset="utf-8"> </head> <body> <h1>Estrutura de um documento  
HTML</h1> <p>Estamos usando os elementos html, head e body.</p> </body>  
</html>
```

- não tem relação com traduzir o conteúdo da página

<a>

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="utf-8">  
<title>Página inicial</title> </head> <body> <h1>Página inicial</h1>  
<p>Você está na página inicial. Veja uma <a href="nomes.html">página com  
nomes de pessoas</a> </p> </body> </html>
```

- referência de um link que possui referência para o link atual → web

- formas de uso
 - associar endereço local/remoto ao **src** de uma
 - texto alternativo que descreve a imagem ao **alt** de uma
 - exibido caso navegador falhe
 - exibido do ponto de vista da acessibilidade
- não tem tag de fechamento
 - elemento auto-fechamento
 - não pode ter conteúdo / elementos alinhados

```

```

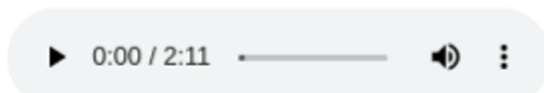
- especificar largura/altura com **width** e **height**
 - modo proporcional → separados
 - modo distorcido → juntos

<audio>

- como usar
 - **controls**: exibição dos controles
 - **source**: nome do arquivo
 - pode ter vários desses
 - indicam várias extensões
 - downloads de 1 a 1 pelo navegador
 - **type**: tipo do arquivo

Minhas músicas

Ouçã algumas músicas que eu gosto.



- outros atributos
 - **autoplay**: toca assim que a página carrega
 - **loop**
 - **muted**

<video>

- atributos
 - iguais de áudio
 - width
 - height
- adição de texto abaixo do elemento para explicar para o usuário que navegador não tem suporte à manipulação de vídeos

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="utf-8">
<title>Meus vídeos</title> </head> <body> <h1>Meus vídeos</h1> <p>Veja
alguns vídeos que gravei.</p> <video controls> <source
src="videos/tocando_violao.mp4" type="video/mp4"> Seu navegador não é
capaz de lidar com vídeos. </video> </body> </html>
```

<iframe>

- vídeo do YTB na sua página

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="utf-8">
<title>Meus vídeos</title> </head> <body> <h1>Meus vídeos</h1> <p>Veja
alguns vídeos que gravei.</p> <video controls> <source
src="videos/tocando_violao.mp4" type="video/mp4"> Seu navegador não é
capaz de lidar com vídeos. </video> <iframe
src="https://www.youtube.com/embed/5mF4pKSi2SU" frameborder="0"></iframe>
</body> </html>
```

- se o link é <https://www.youtube.com/watch?v=5mF4pKSi2SU> → <https://www.youtube.com/embed/5mF4pKSi2SU>

Caracteres especiais

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <title>Caracteres
especiais</title> <meta charset="utf-8"> </head> <body> <p> Um "e
comercial": &amp; </p> <p> Um símbolo de menor: &lt; </p> <p> Um símbolo
de maior: &gt; </p> <p> Raiz quadrada: &#8730; </p> <p> Portanto: &there4;
</p> <p> Copyright: &copy; </p> <p> Marca registrada comercial: &reg; </p>
</body> </html>
```

Elementos Estruturais

- uso
 - organizar outros elementos
 - associar **semântica** à região da página
 - agrupar elementos para aplicar conjunto de regras CSS

<div> e

- não tem significado / semântica
 - ruim para leitores do código

- diferença
 - div: toma tela inteira horizontalmente (`display=block`)
 - span: ocupa somente espaço necessário para seu conteúdo (`display=inline`)

Classes e IDs

- tornar significado mais claro
- problemas
 - seguem convenção própria da equipe
 - idioma específico

Elementos estruturais com semântica

- sem efeito visual, apenas no código
 - mesma funcionalidade de uma `<div>`
- exemplos
 - `<nav>` : barra de navegação
 - `<main>` : conteúdo dominante do `<body>`
 - `<header>` : cabeçalho principal
 - `<section>` : grupo temático de conteúdo
 - `<article>` : exemplos abaixo
 - post de um fórum
 - newspaper
 - blog
 - product card
 - `<footer>` : rodapé para elemento ancestral mais próximo / elemento raiz da seção / página
 - `<aside>` : porção do documento indiretamente relacionada com o core do documento
 - barras laterais

`<form>`

- envio de dados de um usuário ao servidor

- controle: filho de um `<form>`

`<input>`

- controle de um forms
- entrada de dados

- **type**: variar tipo de valores que usuário irá inserir
 - text
 - email
 - navegador validará se é um email válido
 - tel
 - teclado "especial" para celular
 - color
 - file
 - radio
 - seleção de um valor dentre um conjunto de opções

```
<input type="radio"> <input type="radio"> <input type="radio">
<input type="radio"> <input type="radio">
```

Estado: ☒ ☒ ☐ ☒ ☐

- para marcar apenas 1

```
<input type="radio" name="veiculo"> <input type="radio"
name="veiculo"> <input type="radio" name="veiculo"> <input
type="radio" name="veiculo"> <input type="radio" name="veiculo">
```

- rotulando

```
<input type="radio" name="veiculo"><label>X1</label> <input
type="radio" name="veiculo"><label>XC60</label> <input
type="radio" name="veiculo"><label>Q5</label> <input type="radio"
name="veiculo"><label>Discovery</label> <input type="radio"
name="veiculo"><label>Taycan</label>
```

- checkbox

```
<fieldset> <legend>De quais cores você gosta?</legend> <input  
id="vermelho" type="checkbox" name="cor"> <label  
for="vermelho">Vermelho</label> <input id="amarelo"  
type="checkbox" name="cor"> <label for="amarelo">Amarelo</label>  
<input id="preto" type="checkbox" name="cor"> <label  
for="preto">Preto</label> </fieldset>
```

<label>

- informa sobre o que o campo de `<input>` se refere
- seu corpo conterá texto a ser exibido
- `for`: vínculo entre label e input
- `id`: valor associado ao atributo

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="UTF-8">  
<title>Forms</title> </head> <body> <form> <label for="primeiroNome">  
Primeiro nome: </label> <input id="primeiroNome" type="text"> <label  
for="sobrenome">Sobrenome:</label> <input id="sobrenome" type="text">  
</form> </body> </html>
```

- associação com input sem id

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="UTF-8">  
<title>Forms</title> </head> <body> <form> <label> Primeiro nome: <input  
id="primeiroNome" type="text"> </label> <label> Sobrenome: <input  
id="sobrenome" type="text"> </label> </form> </body> </html>
```

- associação do `label` ao `input` com ID

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="utf-8">
<title>Opções</title> </head> <body> 21 <label for="estado">Estado:
</label> <select id="estado"> <option>São Paulo</option> <option>Rio de
Janeiro</option> <option>Minas Gerais</option> <option>Espírito
Santo</option> </select> <input id="X1" type="radio" name="veiculo">
<label for="X1">X1</label> <input id="XC60" type="radio" name="veiculo">
<label for="XC60">XC60</label> <input id="Q5" type="radio" name="veiculo">
<label for="Q5">Q5</label> <input id="Discovery" type="radio"
name="veiculo"> <label for="Discovery">Discovery</label> <input
id="Taycan" type="radio" name="veiculo"> <label
for="Taycan">Taycan</label> </body> </html>
```

Envio de um form

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="UTF-8">
<title>Forms</title> </head> <body> <form> <label> Primeiro nome: <input
id="primeiroNome" type="text"> </label> <label> Sobrenome: <input
id="sobrenome" type="text"> </label> <input type="submit" value="Enviar">
</form> </body> </html>
```

<textarea>

- quando usuário irá inserir texto potencialmente grande
- `cols` e `rows` para especificar largura / altura da caixa

```
<label for="mensagem">Sua mensagem:</label> <textarea id="mensagem"
cols="35" rows="10"></textarea>
```

<select>

- escolha de um valor dentro de uma coleção de opções

```
<label for="estado">Estado:</label> <select id="estado"> <option>São
Paulo</option> <option>Rio de Janeiro</option> <option>Minas
Gerais</option> <option>Espírito Santo</option> </select>
```


Estado: São Paulo ▼

- São Paulo
- Rio de Janeiro
- Minas Gerais
- Espírito Santo

<fieldset>

- agrupamento com semântica dos inputs de um forms
- `<legend>`: especifica uma legenda para o corpo da form

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="utf-8">
<title>Opções</title> </head> <body> <label for="estado">Estado:</label>
<select id="estado"> <option>São Paulo</option> <option>Rio de
Janeiro</option> <option>Minas Gerais</option> <option>Espírito
Santo</option> </select> <fieldset> <legend>Escolha um veículo</legend>
<input id="X1" type="radio" name="veiculo"> <label for="X1">X1</label>
<input id="XC60" type="radio" name="veiculo"> <label
for="XC60">XC60</label> <input id="Q5" type="radio" name="veiculo">
<label for="Q5">Q5</label> <input id="Discovery" type="radio"
name="veiculo"> <label for="Discovery">Discovery</label> <input
id="Taycan" type="radio" name="veiculo"> <label
for="Taycan">Taycan</label> </fieldset> </body> </html>
```

<table>

- `<tr>`: table row
- `<td>`: table data
- `<th>`: table header

```
<table> <!-- cabeçalho --> <thead> <!-- semântico apenas --> <tr>
<th>RA</th> <th>Nome</th> <th>Sobrenome</th> <th>Média</th> </tr> </thead>
<tbody> <!-- semântico apenas --> <!-- primeira linha --> <tr>
<td>123456</td> <td>João</td> <td>Santos</td> <td>10</td> </tr> <!--
segunda linha --> <tr> <td>447744</td> <td>Maria</td> <td>Silva</td>
<td>9</td> </tr> </tbody> <tfoot> <!-- semântico apenas --> <tr> <td>-
</td> <td>-</td> <td>-</td> <td>9.5</td> </tr> </tfoot> </table>
```

- **Tableless**
 - usar `<table>` para organizar layout da página
 - porém, deve ser usado apenas para exibição de dados
 - layout feito por CSS
 - portanto, *tableless* é o Web Design que sugere não utilizar tabelas como layout para estrutura HTML

WAI-ARIA

- especificação que fornece coleção de mecanismos para fornecer acessibilidade no código
- “Como se fosse o CSS para pessoas deficientes visuais”
- não tem impacto visual nenhum
- uso do atributo `role` e `aria-label` nos exemplos abaixo
- princípios
 1. uma `role` é uma promessa
 - ex: prometemos que essa `<div>` desempenha o papel de botão

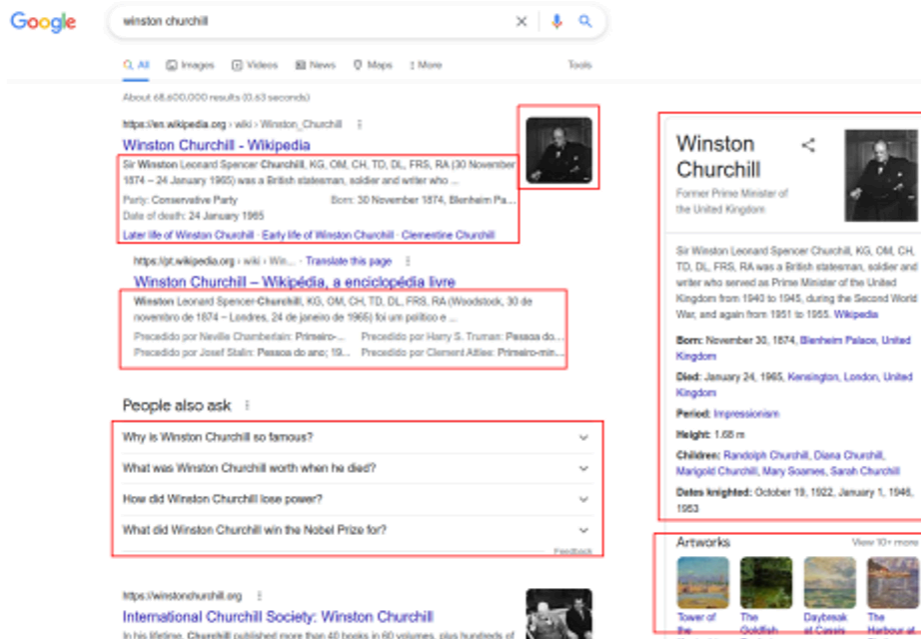
```
<div role="button">Place Order</div>
```

2. ARIA pode ocultar e aprimorar

- ex: aprimorando o que significa um `“menuitem”` a partir do atributo `aria-label`

```
<a role="menuitem">Assistive tech users perceive this element as  
an item in a menu, not a link.</a> <a aria-label="Assistive tech  
users can only perceive the contents of this aria-label, not the  
link text">Link Text</a>
```

Microdados



- especificar informações que os buscadores exibam como microdados
- <https://schema.org/>
- página de exemplo
- `itemscope` → identifica que se quer ter informações sobre uma tag
- `itemtype` → identifica qual item é o referenciado pelo `itemscope`
- `itemprop` → identifica propriedades dentro do `itemscope`
 - é feita dentro das tags filhas da tag referenciada pelo `itemscope`

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="utf-8">
<title>Microdados</title> </head> <body> <div> <article> <h1> A Star is
Born </h1> <p> A musician helps a young singer find fame as age and
alcoholism send his own career into a downward spiral. </p> <div>  </div>
<footer> Diretor: Bradley Cooper. January 5, 1975.</span> </footer>
</article> </div> </body> </html>
```

- o `article` representa um filme, e observando no *schema.org*, tem-se:

```
<div> <article itemscope itemtype="https://schema.org/Movie">
```

- página do tipo `Movie` tem certas propriedades:

```
<article itemscope itemtype="https://schema.org/Movie"> <h1
itemprop="name"> A Star is Born </h1> <p itemprop="abstract"> A musician
helps a young singer find fame as age and alcoholism send his own career
into a downward spiral. </p> <div>  </div> <footer
itemprop="director" itemscope itemtype="https://schema.org/Person">
Diretor: Bradley Cooper. January 5, 1975.</span> </footer> </article>
```

Aula 03 - CSS

CSS Inline

- atributo do elemento que configura o CSS para ele

```
<p style="color: blue">Temos cursos nas mais diversas áreas.</p>
```

- problemas
 - não é possível reutilizar definições
 - regras CSS ficam misturadas com código HTML

Elemento <style>

- uso de seletores CSS

```
<style> /* selecionamos os elementos h1 aqui */ h1 { color: green; } /*
aqui selecionamos os elementos p */ p { color: blue; } </style>
```

- herança de regras
- problemas
 - reusabilidade em outros códigos

Separando em um arquivo

- importando arquivo no *html*

```
<!DOCTYPE html> <html lang="pt-BR"> <head> <meta charset="utf-8">
<title>Cursos</title> <link rel="stylesheet" href="css/styles.css">
</head>
```

Seletores

- especificar elementos que utilizarão a regra CSS
 - tipo do elemento
 - classe (`.`)
 - um ou mais nomes que ficam associados a classes
 - ID (`#`)
- é possível especificar um elemento de certo tipo e com certo ID

```
/* somente elementos do tipo p que tenham id igual a descrição */
p#descricao { color: blue; }
```

- selecionando por classe

```
.promocao { text-align: center; background-color: lightgreen; }
```

- selecionando por tipo e classe

```
p.promocao { text-align: center; background-color: lightgreen; }
```

- utilizando múltiplas classes

```
<p class="nova promocao"> Venha conversar conosco e conhecer nossas
ofertas de <strong>bolsas de estudos</strong>. </p>
```

- selecionando por tipo, classe e id

```
/* todo p que tenha a classe final-pagina e cujo id seja rodape */
p.final-pagina#rodape{ background-color: lightgray; text-align: center; }
```

- agrupando seletores (,)

```
/* todo curso tem essa cor de fundo */ .curso { background-color:
aliceblue; } /* veja a cor de fundo repetida para que possamos especificar
uma regra não aplicável aos demais cursos */ .novo-curso{ background-
color: aliceblue; color: red; } /* unindo a característica do background-
color para ambos os seletores */ /* separamos os seletores usando uma
vírgula. Ela faz um papel de "ou lógico */ .curso , .novo-curso {
background-color: aliceblue; } /* agora não precisamos mais repetir a cor
de fundo */ .novo-curso{ color: red; }
```

```
.curso, .novo-curso, #contato { /* padding: espaço entre o conteúdo e a
borda. Veremos mais sobre isso adiante */ padding: 12px; /* estudaremos
mais sobre bordas também */ border: 1px solid lightgray; }
```

- seletores descendentes (espaço)

```
/* esse é um seletor descendente, representado pelo espaço em branco.
pegamos todos os p's que sejam filhos diretos ou indiretos de um article.
*/ article p{ padding: 12px; background-color: lightgray; } /* todo p que
tenha a classe promoção e que seja descendente de um elemento cujo id seja
principal */ #principal p.promocao { padding: 12px; background-color:
aliceblue; }
```

- filhos diretos (>)

```
/* todos os p's que sejam filhos diretos de article. O p que é filho de
section não é selecionado, portanto. */ article > p { border: 1px solid
black } /* todo p que tenha a classe coordenadores, que seja filho direto
de um section, que seja descendente de um article, que seja filho direto
```