

# CSS - Parte II

Labenu\_



# O que vamos ver hoje?

- Debug
- Combinadores
- Seletores Especiais
- Centralização Horizontal
- Propriedade position
  - static
  - relative
  - fixed
  - absolute
  - sticky



# Debugger

Labenu\_

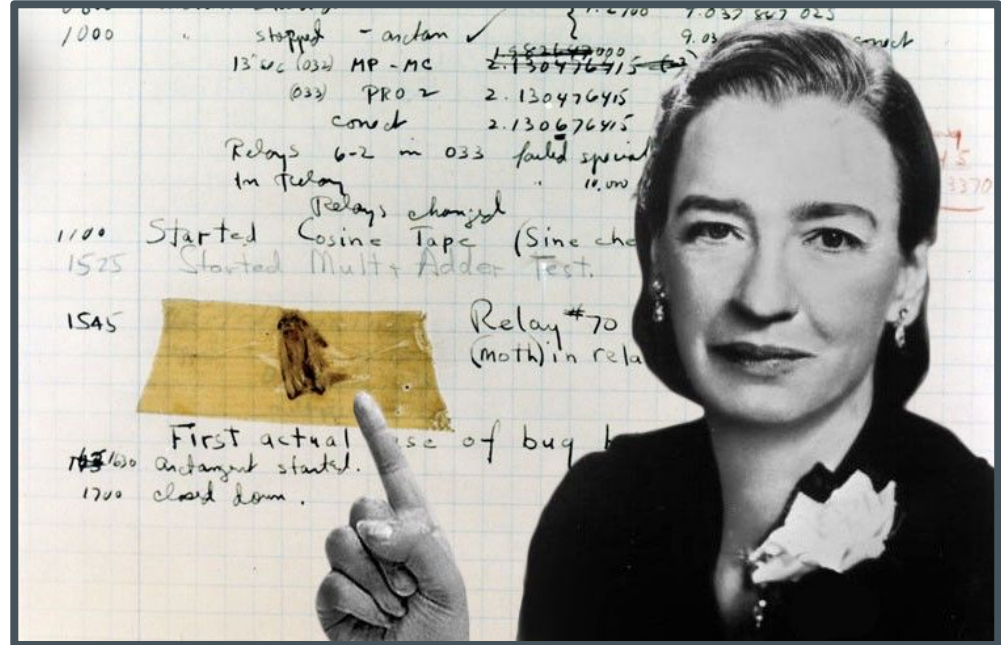


# Debugar

- **Bug** é o termo que utilizamos quando alguma funcionalidade do nosso sistema está com algum problema/defeito
- **Debugar** é um método de **inspecionar** o nosso código para entender porque o problema está acontecendo



# Debugger



# Debugar

- Para debugar os nossos sites, vamos usar o **Chrome Dev Tools**
- *"Conjunto de funcionalidades de autoria e depuração de Web incorporado ao Google Chrome. Use o DevTools para iterar, depurar e criar o perfil do seu site."*
- Hoje veremos 3 abas:
  - **Elements, Console e Application**



# Debugar

- **Elements**

- Permite que **vejamos** o HTML e o CSS e o que cada parte **representa** na tela
- Conseguimos **alterar** a estilização do **CSS** (e, assim, fazer uns testes)
- Ver a estilização **computada**

Vamos ver na prática! 



# Aba Elements

DevTools - www.google.com.br/search?q=asdasd&aq=asdasd&aqs=chrome..69j57j69j60j3j69j65j69j60j315j0j7&sourceid=chrome&ie=UTF-8

Elements Sources Network Performance Console Application Memory Security Audits Redux JavaScript Profiler

```
<!doctype html>
<html itemscope itemtype="http://schema.org/SearchResultsPage" lang="pt-BR">
  <head></head>
  <body class="srp tbo vasq" marginheight="3" topmargin="3" id="gsr">
    <div id="cst"></div>
    <noscript><style>.nojs{visibility:visible}</style></noscript>
    <div></div>
    <noscript></noscript>
    <style></style>
    <div jscontroller="Zy0Bae" jsaction="rcu06b:npT2md" class="jsrp mdm" id="searchform"></div>
    <div class="sfbgx"></div>
    <div id="gac_scont"></div>
    <div class="spch s2fp-h" style="display:none" id="spch"></div>
    <div id="main"></div>
    <script nonce="ab0y7pJ2b/KmB0IR1MADw=="></script>
    <div id="reviewDialog" data-async-context="async_id_prefix:" data-jl="up" data-async-type="reviewDialog" data-async-context-required="async_id_prefix" class="y yp"></div>
    <script nonce="ab0y7pJ2b/KmB0IR1MADw==">window.gbar66gbar.up66gbar.up.tp66gbar.up.tp();</script>
    <script nonce="ab0y7pJ2b/KmB0IR1MADw=="></script>
    <div class="gb_wa"></div>
    <style></style>
    <script src="/js/kxjs.s.pt_BR.XzpWdSRMASI.0/m=Fkg7bd.HcFEGB.IvIue.MC8mtf.0F7_AuDgKrcrCB80I1DdMBA0/rt/d/1/dg=2/rs=ACT980FN9w-69Wk.BMgerJ6grR46VZSmug"></script>
    <script src="/js/kxjs.s.pt_BR.XzpWdSRMASI.0/am=BgDAit17A8_LwBwAUDQKrcrCB80I1DdMBA0.ch.tl.v.s.xz7Ccd.gik19b.aam1I.XjCeUc.F8FRnd.fEVMIc.Ix2Mcc.NBZ7u7xjs=s1" async"></script>
    <script src="/js/kxjs.s.pt_BR.XzpWdSRMASI.0/am=BgDAit17A8_LwBwAUDQKrcrCB80I1DdMBA0.ch.tl.v.s.xz7Ccd.gik19b.aam1I.XjCeUc.F8FRnd.fEVMIc.Ix2Mcc.NBZ7u7xjs=s2" async gapL_processed="true"></script>
    <iframe src="https://clients5.google.com/pagead/drt/dn/" aria-hidden="true" style="display: none !important;" hidden></iframe>
  </body>
</html>
```

html body>gsr.srp.tbo.vasq

Styles Computed Event Listeners DOM Breakpoints Properties Accessibility

Filter

element.style {

```
}
body {
  color: #222;
}
.g, body, html, input, .std, h1 {
  font-family: arial,sans-serif;
  font-size: small;
}
body {
  background-color: #fff;
}
body {
  color: #000;
  margin: 0;
}
body[Attributes Style] {
  margin-bottom: 3px;
  margin-top: 3px;
}
body {
  display: block;
  margin: 0px;
}
Inherited from html
.g, body, html, input, .std, h1 {
  font-family: arial,sans-serif;
  font-size: small;
}
```

margin

border

padding

1397 x 2023.020





# Debugar

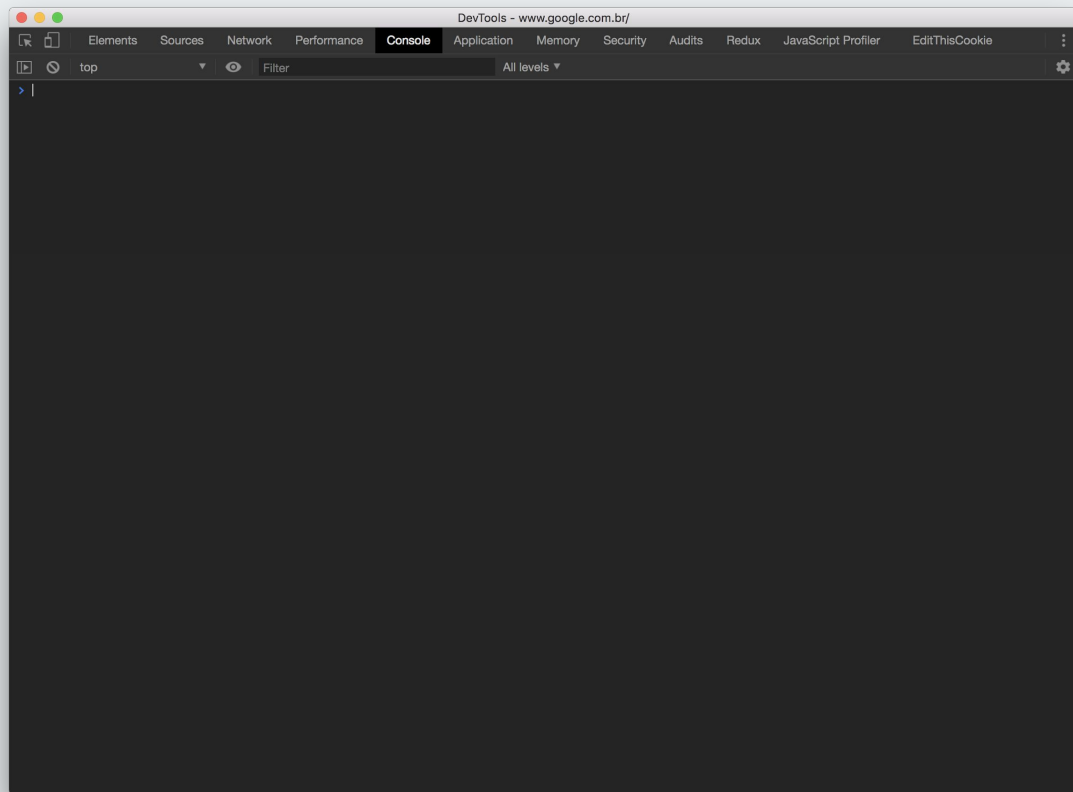
- **Console**

- Permite que vejamos as **saídas** e os **erros** de código em Javascript
- Além disso, é possível **rodar** pequenos códigos nele

Vamos ver na prática! 



# Aba Console



# Debugar

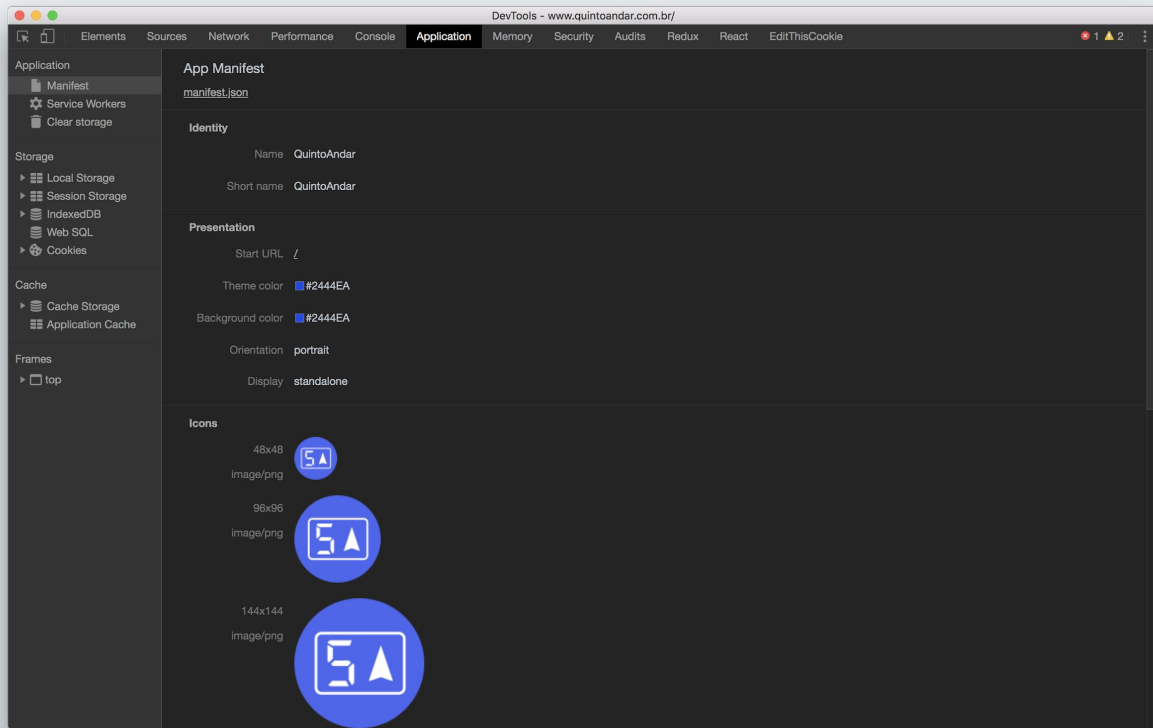
- **Application**

- Permite que vejamos alguns arquivos de configuração
- Ver e limpar dados guardados

Vamos ver na prática! 



# Aba Application



# Voltando ao CSS...

Labenu\_



# Combinadores

Labenu\_



# Combinadores

- Combinadores (**combinators**) são estruturas da sintaxe do CSS que permitem **combinar os seletores**. Hoje, veremos:
  - Múltiplas Condições
  - Seleção Múltipla
  - Descendentes
  - Filhos



# Múltiplas Condições

- Podemos especificar ainda mais o nosso seletor
- Por exemplo: "quero que todos os **h1** com classe **texto** tenham tais características"

```
h1.texto {  
  color: green;  
  border: 1px dotted black;  
}
```

Vamos ver na prática! 





# Seleção Múltipla 🧑 🧑

- Podemos aplicar a **mesma regra** para mais de um seletor, separando-os por **vírgula**
- Abaixo indica um código de CSS que será aplicado em todos os os elementos h1, h2 e h3

```
h1, h2, h3 {  
  color: green;  
  background-color: orange;  
}
```

Vamos ver na prática! 🔬



# Filhos



- Conseguimos especificar o seletor para que ele só atinja os **filhos diretos de um elemento específico**
- Usamos o operador `>`, desta forma:

```
div > p {  
    color: green;  
}
```

Vamos ver na prática! 



# Descendência

- Além de pedir só os filhos, podemos pedir para estilizar **todos os tipos** de um determinado elemento que estiverem **dentro** de outro
- Para isso, separamos os dois por um espaço

```
div p {  
    color: green;  
}
```

Vamos ver na prática! 





# Exercício 1

```
<h1>Sou um título GRANDÃO</h1>
<h2>Sou um título GRANDE</h2>
<h3>Sou um título MÉDIO</h3>
<h4>Sou um título PEQUENO</h4>
<h5>Sou um título PEQUENINO</h5>
<h6>Sou um título MINÚSCULO</h6>
<p> Sou um parágrafo </p>
<h2 class="texto-com-background">
| Sou um título GRANDE
</h2>
<h2
| class="texto-com-background"
| id="texto-com-outra-cor">
| Sou um título GRANDE
</h2>
```

Sou um título GRANDÃO

Sou um título GRANDE

Sou um título MÉDIO

Sou um título PEQUENO

Sou um título PEQUENINO

Sou um título MINÚSCULO

Sou um parágrafo

Sou um título GRANDE

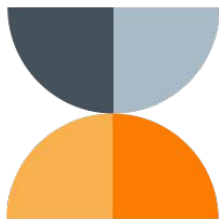
Sou um título GRANDE



# Pausa para relaxar 🧘

10 min

- Combinadores
  - Múltiplas Condições
  - Seleção Múltipla
  - Descendência
  - Filho



# Seletores Especiais

Labenu\_



# Seletores Especiais 👑

- Existem alguns seletores que tem funcionalidades e **responsabilidades especiais**
- Vamos falar de dois especificamente:
  - O seletor \*
  - Pseudo-classes



# Seletores \*



- Permite que sejam atribuídos um conjunto de regras de estilização para **todos** os elementos

```
* {  
    margin: 0;  
    padding: 0;  
}
```

Vamos ver na prática! 





# Seletores \*

- Os navegadores tem alguns **estilos padrão** para os elementos HTML
- Utilizamos o seletor \* para **remover esta estilização padrão**, permitindo que mexamos nos elementos crus



# Seletores de Pseudo-Classes

- São seletores que permitem modificar a estilização do elemento dependendo de **estados especiais**
  - :active
  - :hover



# Seletores de Pseudo-Classes

- **:active** é a pseudo-classe que representa quando o elemento é ativo pelo usuário (normalmente, pelo click)
- **:hover** é a pseudo-classe que representa quando o mouse está por cima do elemento em questão

Vamos ver na prática! 



# Centralização Horizontal

Labenu\_



# Centralização

- **text-align**
  - Permite alinhar os descendentes à partir do elemento ascendente
  - **Funciona para:** texto, imagens e elementos **inline**
  - **Opções:**
    - center, right, left
    - start, end (sentidos da escrita)

Vamos ver na prática! 



# Centralização

- **margin**
  - Para elementos **block** com **largura definida**, podemos usar a propriedade margin
  - Dizemos para ele completar as margens da esquerda e da direita com o valor **auto**

Vamos ver na prática! 





## Exercício 2

- Crie um botão que, ao passar o mouse por cima, fique vermelho com as letras brancas
- Ao ser clicado, seu botão deve ficar rosa com as letras vermelhas
- Centralize o botão horizontalmente



# Pausa para relaxar 🥱

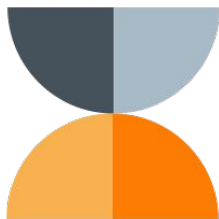
5 min

- **Seletores especiais**

- \*
- Pseudo-Classes
  - :active
  - :hover

- **Centralização horizontal**

- Elementos inline:
  - **text-align**(textos, imagens, etc)
- Elementos block-level:
  - **width** fixo
  - utilizar **margin: 0 auto**





# Position

Labenu\_



# Position

- **position:**
  - Permite indicar o método de posicionamento do elemento
    - static
    - relative
    - fixed
    - absolute
    - sticky



# Position

- **position: static**

- Posição padrão do CSS para todos os elementos
- Segue o fluxo normal de posicionamento de acordo com a ordem dos elementos no html
- Se no html há uma div e logo abaixo outra, elas aparecerão uma abaixo da outra na tela
- Não conseguimos usar as propriedades ~~top~~, ~~bottom~~, ~~left~~ e ~~right~~

**Vamos ver na prática!** 



# Position

- **position: relative**

- Continua seguindo o fluxo normal de posicionamento, porém podemos utilizar as propriedades:
  - top
  - bottom
  - left
  - right
- Posiciona o elemento em relação à posição natural dele (static)

**Vamos ver na prática!** 



# Position

- **position: absolute**

- Retira o elemento do fluxo natural de posicionamento
- Posiciona o elemento em relação à outro elemento *relative* que o englobe
- Utilizamos as propriedades:
  - top
  - bottom
  - left
  - right

Vamos ver na prática! 



# Position

- **position: fixed**
  - Posiciona o elemento em relação à tela toda (ao viewport)
  - Utilizamos as propriedades:
    - top
    - bottom
    - left
    - right

Vamos ver na prática! 



# Position

- **position: sticky**
  - Posiciona o elemento de tal forma que ele fique scrollando até chegar em uma posição. Ao chegar nela, fica "grudado")
  - Para indicar onde ele gruda, usamos as seguintes propriedades:
    - top, bottom, left e right

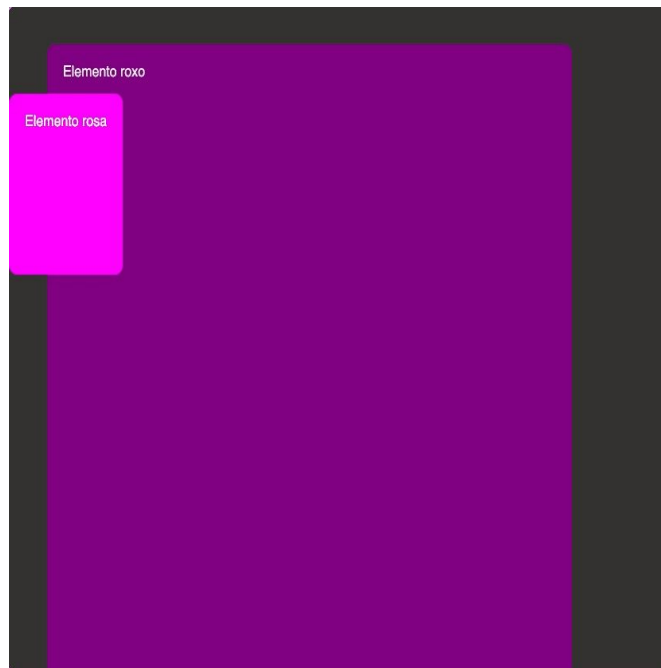
Vamos ver na prática! 



- position: sticky



- position: fixed



Vamos ver na prática! 

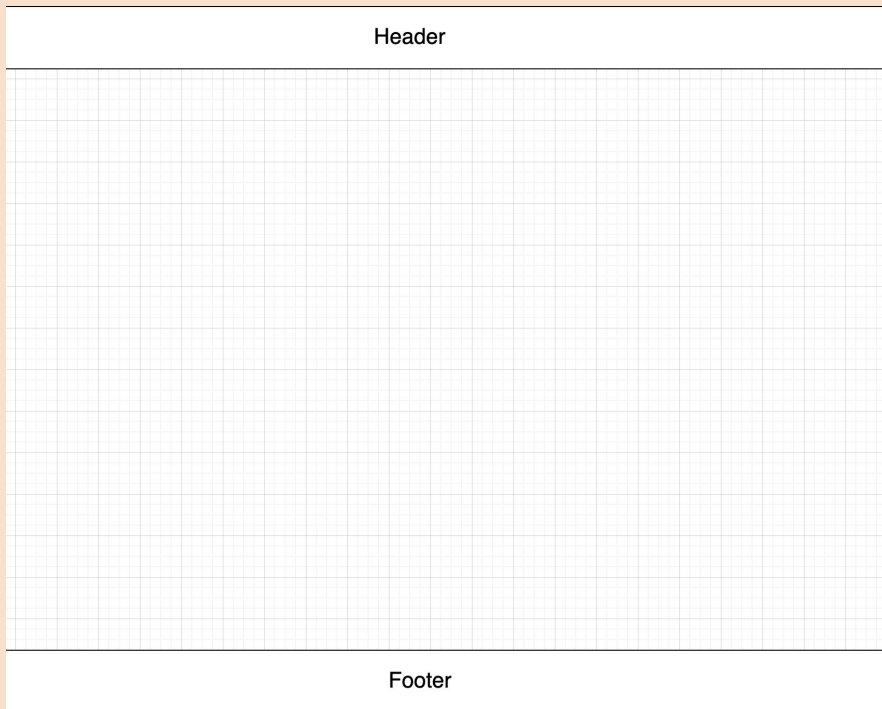






## Exercício 3

- Recriar utilizando HTML e CSS a tela mostrada abaixo



Obs: O header deve ser fixo.  
Ao scrollar na página ele  
deve se manter visível



# Resumo

Labenu\_



# Resumo

- **Combinadores**

- Múltiplas Condições
- Seleção Múltipla
- Descendência
- Filho



# Resumo

- **Seletores especiais**
  - \*
  - Pseudo-Classes
    - :active
    - :hover



# Resumo

- **text-align** permite alinhar textos, imagens e elementos inline **horizontalmente**
- Para centralizar elementos **block-level**, nós devemos colocar um :
  - **width** fixo
  - utilizar **margin-left: auto; margin-right: auto**



# Resumo

- **position** permite trocar o método de posicionamento de um elemento:
  - **fixed**
  - **relative**
  - **absolute**
  - **sticky**
  - **static**
- Utilizamos as propriedades: **top**, **right**, **bottom**, **left** para realizar o posicionamento



# Dúvidas? 🤔

Labenu\_





Obrigado(a)!