**Front-end**

* **Programação WEB**
* O que é programação?
* Ensinar o computador
* Linguagem de programação:
  + Javascript
* Configurar o ambiente de desenvolvimento:
  + Ferramentas
    - Chrome – Navegador
    - VSCode – Plataforma para “codar”
* **Front-end e Back-end**
* Imagina um restaurante:
  + Cliente chama, servidor escuta e reponde
* Como é feita essa comunicação?

🡺 A comunicação entre Empresa/Cliente para troca de dados ocorre através da:

* HTTP-Hyper Text Tranfer Protocol
* URL-Universal Resource Location
* Tecnologias do Front-end
  + HTML
  + CSS
  + Javascript
* Tecnologias do Back-end
  + Veremos nos próximos episódios
* **Configurar o ambiente de desenvolvimento**
* Ferramentas
  + Chrome-Navegador
  + VSCode-plataforma

**O Projeto**

* [Layout](https://www.notion.so/Layout-Happy-Discovery-0517ebde44c24f1b8b23b10b576c4d9f)
* Figma
* Arquivos
* [images.zip](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7b06d2e7-527a-44e2-bfb3-862782837e57/images.zip)

**🏄🏽‍♂️ Introdução às tecnologias**

* HTML
* tags
  + head
    - * meta
      * title
      * link
      * style
  + body
    - * header
        + img
        + div

strong

p

* + - * main
        + h1
        + section

p

a

img

* CSS
* O que é?
  + - Cascading Style Sheets
    - Apresentação visual para o cliente
    - Estilo para o HTML
* Fundamentos
  + - Sintaxe / Anatomia
      * Regra

/\* selector body \*/

body {

/\* declaration \*/

/\* property: value; \*/

color: red;

} /\* end of rule \*/

* + - Cascata e Especificidade
    - Valores: named, functions, numeric
* Responsivo
  + - Mobile First
    - Unidades de medida: rem / em
* Box Model
  + - tamanho do conteúdo (width / height )
    - preenchimentos
    - bordas
    - espaçamentos
    - cores
      * fundo
      * sombras
      * opacidade
* Layout
  + - Posicionamentos
      * displays
        + regras fora da caixa ( block / inline )

display: block

elements: div / h1 / p

display: inline

elements: a / span / strong

* + - * + regras para dentro da caixa ( flex / grid )
    - position: relative / absolute
    - alinhamentos
* Textos/Fontes
  + - tamanho
    - tipo
    - espaçamentos
    - alinhamentos
* Animações e Transições
* Animações CSS tornam possível animar transições de um estilo CSS para outro. Animações se consistem de dois componentes: um estilo descrevendo a animação e um set de keyframes que indicam o estado final e inicial do estilo CSS da animação, bem como possíveis waypoints intermediários ao longo do caminho.
  + **Exemplo:**

p {

animation-duration: 3s;

animation-name: slidein;

}

@keyframes slidein {

from {

margin-left: 100%;

width: 300%

}

to {

margin-left: 0%;

width: 100%;

}

}

* Javascript
* Variáveis
* Tipos de dados
* Objetos
* Funções
* Condições
* Repetições