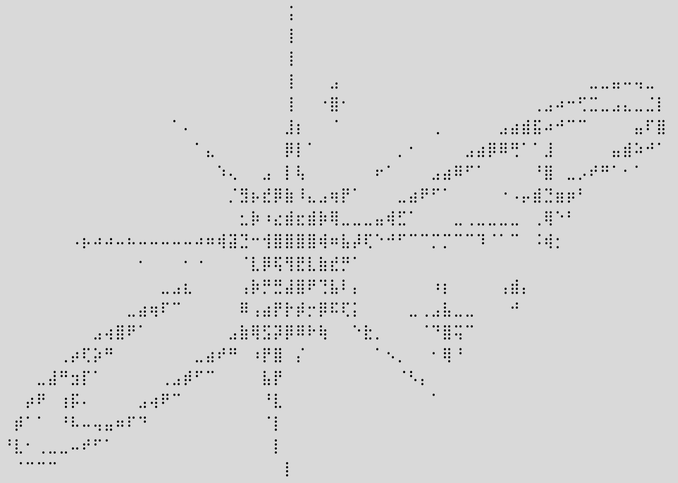
♡ ╮(˘◡˘)╭ ♡

javascript

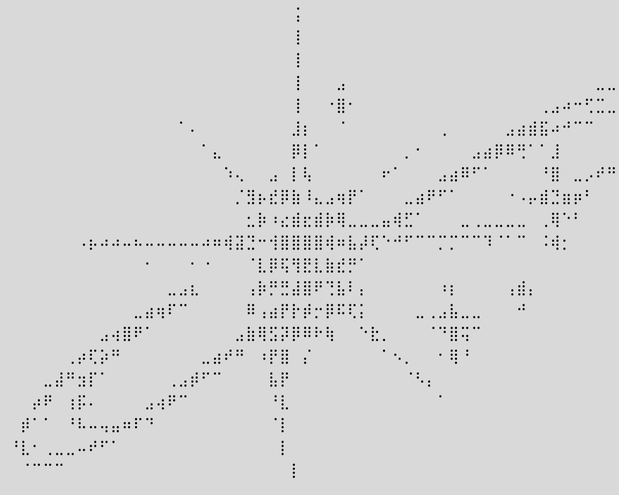
javascript

agora que compreendemos os fundamentos da web, podemos começar a pensar em javascript :-)



js permite implementar itens mais complexos nas nossas páginas. serve para adicionar funcionalidades interativas e outros conteúdos dinâmicos como por exemplo formúlarios, animaçÕes, etc.

para perceber melhor como js funciona, este slideshow sugere um exercício mas antes, vamos falar de alguns conceitos básicos :-)



javascript - funçõe‌s / functions

funçõe‌s sã‌o um dos blocos fundamentais do js. uma funçã‌o é essencialmente um conjunto de instruçõe‌s que executa uma tarefa ou calcula um valor — e que podemos reutilizar sempre que precisarmos. Neste exemplo vamos escrever uma funçã‌o simples: um conjunto de tarefas que queremos que seja executado sempre que essa funçã‌o for chamada.

Mais funçoes em w3schools e mdn docs.

javascript - variáveis / variables

variáveis, escritas como 'var', 'let' ou 'const', são contentores para guardar informação. neste exemplo vamos usar uma variável para criar uma forma abreviada de referenciar uma sequência de texto mais longa — neste caso um 'método' — que será usada para seleccionar um elemento no nosso html.

mais variáveis em w3schools e mdn docs.

javascript - eventos / events

um evento é algo que o browser faz, ou algo que o utilizador faz. os eventos acontecem constantemente em segundo plano à medida que o utilizador interage com a página, e podemos usar javascript para reagir a esses eventos. neste exemplo vamos usar o evento 'onclick' para activar a nossa função. como o nome sugere, este evento é desencadeado quando o utilizador clica num elemento específico.

mais eventos em w3schools.

javascript - exercício

vamos começar por escrever javascript directamente dentro do nosso documento html. podemos colocá-lo em qualquer sítio, mas é recomendado colocá-lo no fundo do elemento body, mesmo antes da tag de fecho. colocar os scripts aqui melhora a velocidade de carregamento da página — se forem encontrados no topo do documento, podem atrasar a renderização. se estamos a adicionar javascript ao nosso documento html, o código tem de estar escrito entre tags script.
copia e cola este script com uma função simples para o fundo do teu elemento body:

```
<script>
function change() {
    console.log('it works!');
}
</script>
```

javascript - exercício

por si só, esta função não vai fazer nada até definirmos como queremos executá-la. para isso, vamos criar um botão com um evento onclick. podemos colocá-lo em qualquer sítio dentro do elemento body:

```
<button onclick=' ' >clica aqui...</button>
```

agora vamos adicionar o nome da função 'change()' ao atributo onclick. não te esqueças dos parênteses!

```
<button onclick=' change()' >clica aqui...</button>
```

isto deve ser suficiente para o javascript começar a funcionar. no entanto, por agora, não vamos ver nada acontecer na página quando clicamos no botão. para confirmar que está a funcionar, vamos abrir as ferramentas de desenvolvimento do browser (botão direito + inspecionar) e navegar para o separador 'console'. a consola é o registo dedicado a js e erros do browser. sempre que estamos a escrever ou a fazer debug de js, é boa ideia ter a consola aberta para confirmar que tudo está a funcionar como esperado.

javascript - exercício

dentro da nossa função temos uma única linha:

```
console.log('it works!');
```

esta linha envia para a consola do browser o texto que estiver dentro dos parênteses, sempre que é executada. se clicares no botão e 'it works!' aparecer na consola, o teu javascript está a funcionar correctamente.
agora que sabemos que funciona, vamos fazer algo mais interessante. vamos adicionar esta linha ao script, por baixo da linha console.log():

```
document.querySelector(' .CLASS' ).classList.toggle(' NEW-CLASS' );
```

há muita coisa a acontecer aqui, mas essencialmente esta linha vai activar e desactivar uma classe num elemento específico sempre que for executada. especificamos o elemento que queremos seleccionar com querySelector() e designamos a classe que queremos alternar com classList.toggle().

javascript - exercício

atenção às maiúsculas e minúsculas! esta forma de escrever chama-se camelCase e é essencial seguir exactamente, caso contrário as funçÕes não vão funcionar correctamente. agora vamos criar um elemento que queremos alterar quando clicamos no botão:

```
<div class=' menu' > Hello World! </div>
```

vamos actualizar a linha do toggle no javascript com a classe da nossa div:

```
document.querySelector(' .menu' ).classList.toggle(' NEW-CLASS' );
```

a seguir, vamos definir o nome da classe que queremos activar no elemento quando o botão é clicado:

```
document.querySelector(' .menu' ).classList.toggle(' active' );
```

por fim, definimos a classe .active no nosso css:

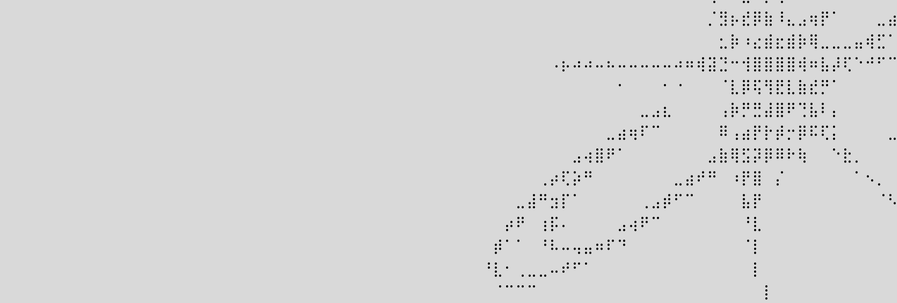
```
.active { background: lightblue; font-size: 150%; }
```

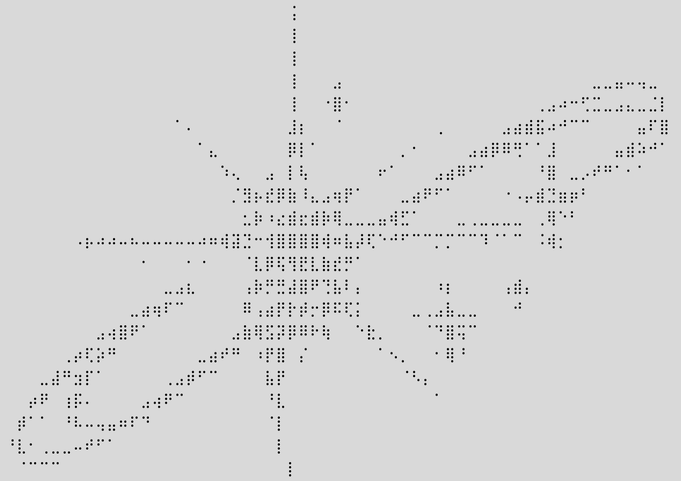
jquery e bibliotecas

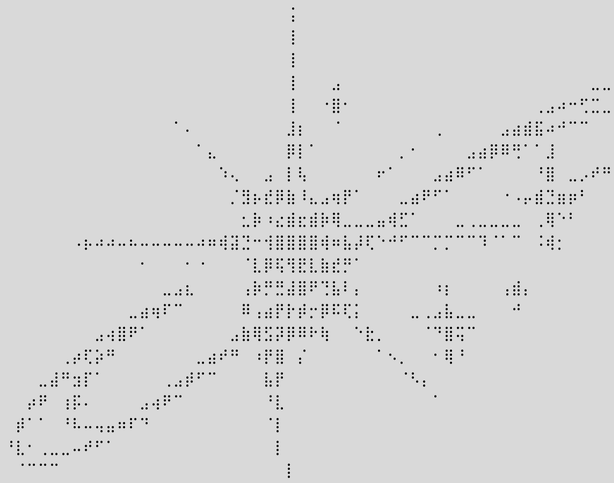
jquery

agora que já temos uma noção básica de javascript, vamos explorar uma biblioteca muito popular chamada jQuery. uma biblioteca é uma colecção de funções, operações e outros fragmentos de código que expandem ou estendem o uso de uma linguagem de programação. o jQuery é uma biblioteca javascript muito utilizada que se define como 'uma biblioteca javascript rápida, pequena e rica em funcionalidades.' essencialmente, simplifica a programação em javascript ao reduzir o tamanho da sintaxe padrão e ao agrupar funções de uso comum em métodos mais curtos e fáceis de usar.

podes ler mais em w3schools jQuery intro.







jquery

primeiro, precisamos de fazer download da versão mais recente 'compressed production build' aqui.
copia o ficheiro .js do jQuery para a pasta do teu website. de seguida, vamos criar uma ligação para ele no nosso documento HTML. vamos colocá-la no fundo do body:

```
<script src=' jquery.js' ></script>
```

jquery

com isto, instalámos o jquery na nossa página e podemos começar a usar as suas funçÕes no nosso javascript.
de seguida, vamos criar um novo ficheiro e chamá-lo algo como 'main.js'. este será o ficheiro .js onde vamos adicionar as nossas próprias funçÕes. funciona de forma semelhante à nossa folha de estilos CSS externa — permite-nos escrever scripts num documento separado que podemos facilmente aplicar a várias páginas se quisermos.
cria uma ligação para ele da mesma forma que ligámos o ficheiro 'jquery.js', mas certifica-te de que fica abaixo da referência ao jQuery. a ordem aqui é importante: o javascript que vamos escrever no nosso ficheiro 'main.js' vai depender da biblioteca jQuery acima dele — e como já sabemos, o documento HTML é lido de cima para baixo, por isso as coisas não vão funcionar se não estiverem na ordem certa.

```
<script src=' main.js' ></script>
```

jquery

agora que temos tudo configurado, vamos começar a experimentar com jQuery. podemos começar com algo semelhante ao exercício anterior: o método toggleClass. vamos ver o exemplo do w3schools aqui.

mais coisas:

w3schools - jQuery Examples

jQuery UI

p5.js

paper.js

three.js

flickity

♡ ╮(˘◡˘)╭ ♡