

Estruturas condicionais

São instruções para realizar determinadas tarefas a partir de uma condição, seja de decisão ou repetição;

Ex.: Um jogo precisa mudar o placar toda vez que um jogador marca pontuação;

if

Podemos usar as palavras reservadas “if” para estabelecer uma condição:

Ex.: var jogador1 = 0;
var jogador2 = 0;

```
if (jogador1 > 0) {  
  console.log('jogador1 marcou ponto');  
}
```

// como ler: se o jogador1 tiver valor maior que 0 ele marcou ponto.

else

No caso de a condição não ser atendida podemos usar o “else” :

→ Ex.: var jogador1 = 1; var jogador2 = 0;

```
if (jogador1 > 0) {  
  console.log('jogador1 marcou ponto');  
} else {  
  console.log('ninguem marcou ponto');  
}
```

else if

Caso haja mais de uma condição usamos o “else if”.

→ Ex.: var jogador1 = 1; var jogador2 = 0;

```
if (jogador1 > 0) {  
  console.log('jogador1 marcou ponto');  
} else if (jogador2 > 0) {  
  console.log('jogador2 marcou ponto');  
} else {  
  console.log('ninguém marcou ponto');  
}
```

ninho de if

Podemos também usar o “if” dentro de um outro “if”, chamamos isso de aninhamento de if 's ou ninho de if 's.

→ Ex.: if (jogador1 = -1) {

```
  if (jogador1 > 0) {  
    console.log('jogador 1 marcou ponto');  
  } else {  
    console.log('ninguém marcou ponto');  
  }  
} else {  
  console.log('jogador inválido');  
}
```

If ternário

Podemos também fazer uma verificação em uma única linha usando o “if” ternário:

→ Ex.: [condição] ? [instrução1] : [instrução2];

```
jogador1 > 0 ? console.log('marcou ponto') : console.log('não  
marcou ponto');
```

// lembre de usar a interrogação “?” e dois pontos “.”

switch/case

O “switch/case” funciona como uma estrutura condicional também;

→ Ex.: switch (\${expressao}) {

```
case 1:
  ${instrucao};
  break;
case 2:
  ${instrução};
  break;
}
```

Estruturas de repetição

→ São estruturas condicionais que repetem uma instrução até atingir determinada condição:

- For;
- For/in;
- For/of;
- While;
- Do/while;

For

Funciona como uma repetição de instrução até que a condição seja falsa:

```
for ([expressaoInicial]; [condicao]; [incremento]) {
  declaracao }

```

Ex.: var array = ['valor1', 'valor2', 'valor3', 'valor4']

```
for (let i = 0; i < array.length; i++) {
  console.log(i);
}
```

For/in

Funciona como uma repetição a partir de uma propriedade:

```
for ([indice] in [objeto ou array]) {  
  declaracao }  
}
```

Ex.: var array = ['valor1', 'valor2', 'valor3', 'valor4']

```
for (i in array) {  
  console.log(i);  
}
```

For/of

Funciona como uma repetição a partir de um valor:

```
for ([indice] of [array]) {  
  declaracao }  
}
```

Ex.: var array = ['valor1', 'valor2', 'valor3', 'valor4']

```
for (i of array) {  
  console.log(i);  
}
```

For/of

O For/of não funciona com objetos pois as propriedades variam, diferentes do índice em um array que sempre serão números inteiros.

Mas e se eu quiser pegar o valor mesmo assim?

```
→ Ex.: for (i of object.propriedade) {  
  console.log(i);  
}
```

// porém cada caractere dentro do valor será
Impresso em linhas separadas.

While

Executa uma instrução “enquanto” determinada condição for verdadeira, a verificação é feita antes da execução;

```
→ Ex.: var a = 0;
    while (a < 10) {
        a++;
        console.log(a);
    }
```

// como ler: enquanto a variável a for menor que 10 ela vai receber mais um e imprimir no console.

Do/while

Executa uma instrução “até que” determinada condição seja falsa, a verificação é feita depois da execução;

```
→ Ex.: var a = 0;
    do {
        a++;
        console.log(a);
    } while (a < 10)
```

// como ler: a variável a vai receber mais um e imprimir no console até que seu valor chegue a 10