

JavaScript

Aula 3



JS

Switch

Esta instrução é bem semelhante com uma estrutura IF, porém é mais eficiente em razão de ser mais simples sua utilização e seu entendimento.

```
switch (variável)
{
    case CONSTANTE:  Valor numérico e string
        comandos; Executa o comando
        break;         Força a parada do código sem ir adiante nos demais cases
    case CONSTANTE2:
        comandos;
        break;
    case default:
        comandos;
        break;
}
```

JS

switch() → significa “desvio”, é a abertura da estrutura e recebe a entrada da variável que será comparada.

Case → significa “caso”, é o espaço onde incluimos cada opção de resposta, que será executada “caso” seja igual ao seu valor de entrada. O case define o ponto de entrada, seguindo sua execução linha-a-linha até o fim, ou até ser interrompido (break).

Break → significa “interrompimento”, comando necessário ao final de cada bloco de linhas da opção case. O break deve ficar entre o início do case e seu término, criando uma estrutura de blocos e definindo um ponto de saída. Se não informado, todas as demais linhas do switch serão executadas.

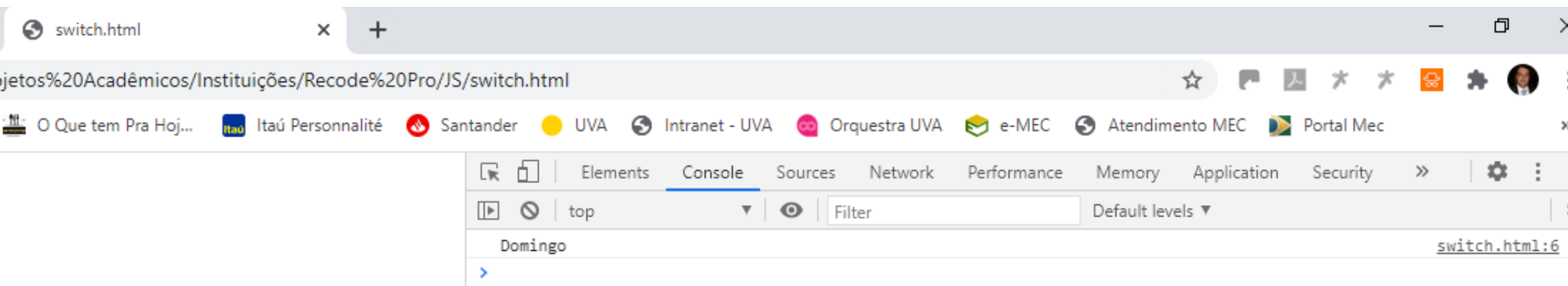
Default → significa “padrão”, é o bloco de código que será executado caso nenhuma opção existente seja a correta.

JS

Exemplo:

```
<script>
let dia = 0;
switch (dia)
{
  case 0:
    console.log("Domingo");
    break;
  case 1:
    console.log("Segunda");
    break;
  case 2:
    console.log("Terça");
    break;
  case 3:
    console.log("Quarta");
    break;
  case 4:
    console.log("Quinta");
    break;
  case 5:
    console.log("Sexta");
    break;
  case 6:
    console.log("Sabado");
    break;
  default:
    console.log("Dia da Semana Invalido");
}
</script>
```

JS



JS

Exemplo:

```
<script>
let cor = "b";
switch (cor)
{
  case "a":
    console.log("azul");
    break;
  case "b":
    console.log("branco");
    break;
  case "c":
    console.log("caqui");
    break;
  case "d":
    console.log("dourado");
    break;
  case "e":
    console.log("esmeralda");
    break;
  case "f":
    console.log("firebrik");
    break;
  case "g":
    console.log("gold");
    break;
  default:
    console.log("Cores Invalidas");
}
</script>
```

Exemplo:

```
<script>
let trimestre = 5;
switch (trimestre)
{
  case 1:
  case 2:
  case 3:
    console.log("primeiro trimestre");
    break;
  case 4:
  case 5:
  case 6:
    console.log("segundo trimestre");
    break;
  default:
    console.log("Trimestre Invalido");
}
</script>
```

Comandos condicionais e repetição

Instrução While

A instrução while realiza uma ação enquanto determinada condição for satisfeita.

Sua sintaxe básica é:

```
while (expressão) {  
    <comando>;  
    <comando>;  
}
```


JS

Veja no exemplo seguinte a utilização do laço while que é repetido por total de 10 vezes:

```
<script>
```

```
    let num = 0;
```

```
    while (num < 10) {
```

```
        console.log("Número: "+num+"<br>");
```

```
        num++;      /* faz o incremento para a variável  
                     num*/
```

```
    }
```

```
</script>
```

JS

The screenshot shows a web browser window with the address bar displaying the URL: `objetos%20Acadêmicos/Instituições/Recode%20Pro/JS/whileconsole.html`. The browser's developer tools are open, and the 'Console' tab is selected. The console shows the output of a JavaScript while loop, displaying numbers from 0 to 9, each followed by a line break (`
`). The source of each log entry is `whileconsole.html:4`.

Browser tabs and bookmarks are visible at the top, including 'whileconsole.html', 'O Que tem Pra Hoj...', 'Itaú Personalité', 'Santander', 'UVA', 'Intranet - UVA', 'Orquestra UVA', 'e-MEC', 'Atendimento MEC', and 'Portal Mec'.

The developer tools interface includes tabs for Elements, Console, Sources, Network, Performance, Memory, Application, and Security. The Console tab shows a list of log entries with a filter input and a 'Default levels' dropdown.

Log Entry	Source
Número: 0 	whileconsole.html:4
Número: 1 	whileconsole.html:4
Número: 2 	whileconsole.html:4
Número: 3 	whileconsole.html:4
Número: 4 	whileconsole.html:4
Número: 5 	whileconsole.html:4
Número: 6 	whileconsole.html:4
Número: 7 	whileconsole.html:4
Número: 8 	whileconsole.html:4
Número: 9 	whileconsole.html:4

JS

Instrução for

A instrução for realiza uma ação até que determinada condição seja satisfeita.

Sua sintaxe básica é:

```
for (início; condição; incremento) {  
    <comando>;  
}
```

O início determina o valor inicial do laço for. Normalmente é 0 ou 1, porém poderá ser especificado qualquer outro valor. O valor especificado é atribuído em uma variável, por exemplo `i=0`, `count=1`.

JS

A condição determina a expressão que irá controlar o número de repetições do laço. Enquanto esta expressão for verdadeira, o laço continuará, caso o laço seja falso, o laço terminará. Por exemplo: `i<20`. Enquanto o valor de `i` for menor que 20, a condição é verdadeira.

O incremento determina como o laço irá contar, de 1 em 1, 2 em 2, 5 em 5, 10 em 10, enfim.

Exemplo: `i++`. Será aumentado o valor da variável `i` a cada repetição. Diferente de todas as outras linguagens, em JavaScript, a instrução `for`, utiliza ponto e vírgula para separar os argumentos ao invés de vírgula.

JS

Vejamos um exemplo prático de utilização do laço for que conta valores de 1 até 10, acrescentando um valor de cada vez:

Exemplo:

```
<script>  
    let num = 0;  
    for (i=1 ; i<=10 ; i++){  
        console.log("número: "+ i + "<br>");  
    }  
</script>
```

JS

