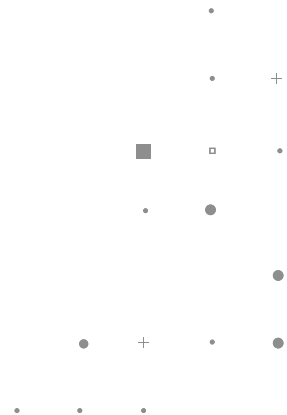




FIAP

GRADUAÇÃO





TDS

FRONT-END DESIGN ENGINEERING

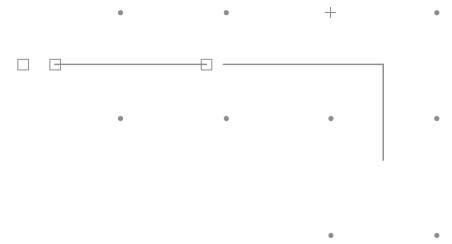
Prof. Luís Carlos

lsilva@fiapcom.br



JavaScript

Datas e Horas





COMO TRABALHAR COM DATAS E HORAS?

Quando precisamos manipular **data e horas no Client Side** (lado do cliente), precisamos utilizar uma classe que facilita bastante esta tarefa.

Date() - é uma classe que tem todos os métodos necessários para trabalhar com datas. Basta chamar os métodos certos para ter o retorno esperado.

```
let data = new Date();  
console.log(data);
```

OBS. Temos que tomar cuidado, pois como estamos trabalhando do lado do cliente os valores de data e horário são os da máquina do nosso usuário.





COMO TRABALHAR COM DATAS?

Temos métodos para extrair cada parte das informações disponíveis nesta classe, assim podemos personalizar a exibição das informações.

getDate() – este método retorna o dia corrente.

```
let data = new Date().getDate()  
console.log(data);
```

OBS. Se quisermos retornar o dia com 2 dígitos, caso menor que dez, temos que formatar a saída.
Por exemplo:

```
let data = new Date().getDate()  
console.log(('0'+data).slice(-2));
```



COMO TRABALHAR COM DATAS?

getMonth() – este método retorna o mês, mas em forma de vetor, ou seja, começa a contagem a partir do zero.

```
let data = new Date().getMonth()+1  
console.log(data);
```

getFullYear() – este método retorna o ano com quatro dígitos.

```
let data = new Date().getFullYear()  
console.log(data);
```



COMO TRABALHAR COM DATAS?

getDay() – este método retorna o dia da semana, mas em forma de vetor, ou seja, começa a contagem a partir do zero. Se quisermos o dia devemos utilizar um array

```
let data = new Date()  
let diasSemana = ['Domingo', 'Segunda', 'Terça', 'Quarta',  
                  'Quinta', 'Sexta', 'Sábado']  
console.log(diasSemana[data.getDay()]);
```



COMO TRABALHAR COM DATAS?

Podemos também modificar estes valores:

setDate() – Seta novos valores na data em dias.

```
let data = new Date()  
console.log(data);  
data.setDate(data.getDate()+5)  
console.log(data);
```




COMO TRABALHAR COM DATAS?

setMonth() – Seta novos valores na data em meses.

```
let data = new Date()  
console.log(data);  
data.setMonth(data.getMonth()+5)  
console.log(data);
```

setFullYear() – Seta novos valores na data em anos.

```
let data = new Date()  
console.log(data);  
data.setFullYear(data.getFullYear()+5)  
console.log(data);
```



COMO TRABALHAR COM DATAS?

toString() – retorna a data no padrão americano.

```
let data = new Date()  
console.log(data.toString());
```

toLocaleDateString() – retorna a data completa no padrão do navegador.

```
let data = new Date()  
console.log(data.toLocaleDateString());
```



COMO TRABALHAR COM DATAS?

Podemos formatar a data completa assim, por exemplo:

```
let data = new Date()
let dia = ('0'+data.getDate()).slice(-2)
let mes = data.getMonth()
let meses = ['Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio',
  'Junho', 'Julho', 'Agosto', 'Setembro', 'Outubro', 'Novembro', 'Dezembro']
let ano = data.getFullYear()
console.log(dia + ' de '+meses[mes]+' de '+ano);
```



COMO TRABALHAR COM HORAS?

getHours() – este método retorna a hora do sistema local.

```
let data = new Date()  
console.log(data.getHours());
```

getMinutes() – este método retorna os minutos do sistema local.

```
let data = new Date()  
console.log(data.getUTCMinutes());
```



COMO TRABALHAR COM HORAS?

getSeconds() – este método retorna os segundos do sistema local.

```
let data = new Date()  
console.log(data.getSeconds());
```



COMO TRABALHAR COM HORAS?

Manipulando as horas também:

setHours() – este método manipula as horas.

```
let data = new Date()  
console.log(data.getHours());  
data.setHours(data.getHours()+3)  
console.log(data.getHours());
```



COMO TRABALHAR COM HORAS?

setMinutes() – este método manipula os minutos.

```
let data = new Date()  
console.log(data.getMinutes());  
data.setMinutes(data.getMinutes()+15)  
console.log(data.getMinutes());
```

setSeconds() – este método manipula os segundos.

```
let data = new Date()  
console.log(data.getSeconds());  
data.setSeconds(data.getSeconds()+20)  
console.log(data.getSeconds());
```



COMO TRABALHAR COM HORAS?

toLocaleString() – este método retorna a data e o horário completo.

```
let data = new Date()  
console.log(data.toLocaleString());
```

OBS. Podemos usar o toLocaleString() para extrair apenas as horas completas.

```
let data = new Date()  
console.log((data.toLocaleString()).slice(-8));
```


JavaScript

Manipulando o Tempo





MANIPULANDO O TEMPO

Nem sempre queremos que a reação aos eventos sejam realizados no exato momento que sejam chamado. Assim podemos usar métodos que nos auxiliam neste controle.

Temos também eventos que precisam ser executados de tempo em tempo, para estes casos também temos uma solução.





MANIPULANDO O TEMPO

setTimeout(função, tempo) – executa a função após a contagem do tempo determinado em milissegundos.

```
function ola(){  
  alert('Olá pessoal, demorei?')  
}  
  
setTimeout(ola, 4000)
```

OBS. O **setTimeout** só executa a função uma vez.



MANIPULANDO O TEMPO

setInterval(função, tempo) – recebe a função e o tempo ,fica repetindo aquela função a cada intervalo.

```
function tempo(){  
    let relógio = document.querySelector('#div1')  
    let hora = new Date()  
    relógio.innerHTML = (hora.toLocaleString()).slice(-8)  
}  
  
setInterval(tempo, 1000)
```

OBS. Já **setInterval** fica executando a função indeterminadamente.



MANIPULANDO O TEMPO

Mas e se for preciso para a repetição da função que está no setInterval()?

clearInterval(variável) – recebe a variável a que o setInterval foi atribuído.

```
<div id="div1">  
  Relógio  
</div>  
<button id="ligar">Iniciar</button>  
<button id="desligar">Parar</button>
```

```
let tmp  
function tempo(){  
  let relógio = document.querySelector('#div1')  
  let hora = new Date()  
  relógio.innerHTML = (hora.toLocaleString()).slice(-8)  
}  
  
document.querySelector('#ligar').addEventListener('click',()=>{  
  tmp = setInterval(tempo,1000)  
})  
  
document.querySelector('#desligar').addEventListener('click',()=>{  
  clearInterval(tmp)  
})
```

DÚVIDAS?

- lsilva@fiap.com.br

“A dúvida é o princípio da sabedoria.”

- Aristóteles



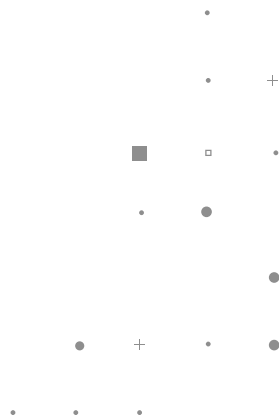


OBRIGADO

FIAP

Copyright © 2020 | Professor (a) Luis Carlos S. Silva

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.





FIAP

