



escola  
britânica de  
artes criativas  
& tecnologia

# Desenvolvedor Full Stack Python

Durante este módulo construímos mais um projeto, fizemos uma landing page para um evento fictício: EBAC Tech Talks.

Você pode consultar o código escrito durante o módulo clicando [aqui.](#)

# Projeto – EBAC Tech Talks

Durante a construção desta landing page conhecemos uma nova função do JavaScript, o **setInterval**, com essa função podemos executar uma função após determinada intervalo.

```
setInterval(FUNCAO, intervalo);
```

# Projeto – EBAC Tech Talks

No primeiro argumento inserimos o código que será executado a cada intervalo e no segundo argumento passamos o tempo em milissegundos.

```
setInterval(function() {  
    console.log("olá");  
}, 1000);  
// A cada 1 segundo teremos a mensagem olá no console
```

# Projeto – EBAC Tech Talks

Para parar um intervalo utilizamos a função `clearInterval`, passando o intervalo como argumento, para isso precisamos armazenar o `setInterval` em uma variável:

```
const contador = setInterval(function() {  
  console.log("olá");  
}, 1000);  
// ....  
clearInterval(contador); // o contador será  
paralisado, para retomar será necessário criar um  
novo setInterval.
```

# Projeto – EBAC Tech Talks

Também conhecemos os recursos de data do JavaScript.

Podemos criar uma data utilizando o comando `new Date()`, a partir disso teremos acesso a funções para recuperar o dia, mês, ano e obter a data em timestamp.

```
const minhaData = new Date();
```

# Projeto – EBAC Tech Talks

Considerando a constante `minhaData` criada na página anterior:

```
minhaData.getTime() // retorna a data atual em timestamp  
minhaData.getDate() // retorna o dia - número  
minhaData.getDay() // retorna o dia, como índice, onde  
                        domingo = 0, segunda = 1  
minhaData.getFullYear() // retorna o ano completo, ex: 2022  
minhaData.getYear() // Retorna o ano porém considera o ano  
2000 como 100, logo 2022 seria 122  
minhaData.getHour() // retorna a hora atual em 24hrs  
minhaData.getMinute() // retorna o minuto atual  
minhaData.getSeconds() // retorna o segundo atual
```

# Projeto – EBAC Tech Talks

Também podemos utilizar o `new Date()` para criar uma data específica, no passado ou futuro:

```
minhaData.getTime() // retorna a data atual em timestamp  
const dataNoPassado = new Date("April 24, 2000 19:00:00")  
const dataNoFuturo = new Date("December 20, 2030 19:00:00")  
// Seguimos a estrutura  
// (mês em inglês dia, ano horário 24hrs)
```