

Objeto II

Objetos Nativos

Os objetos nativos são objetos que vem da própria linguagem.

A exemplo temos alguns bem conhecidos como:

```
Math  
String  
Boolean  
Number
```

Um exemplo de utilização desses objetos é quando criamos variáveis. Geralmente utilizamos da notação literal, como demonstrado a baixo:

```
var nome = 'João';
```

Mas também podemos utilizar da maneira formal, utilizando o nosso objeto String

```
var nome = new String('João');
```

Ambos terão o mesmo valor, mas um é um tipo primitivo e outro é objeto.

A diferença que o objeto irá possuir atributos e métodos que pode ser utilizado para inúmeras situações.

Como exemplo o atributo length que retorna a quantidade de carácter em uma string.

```
var nome = new String('João')  
nome.length //Retornará 4
```

Objeto do ambiente de hospedagem

Esse objeto diz respeito a onde o código está rodando, isso significa que alguns objetos podem variar dependendo do local onde o código está em execução.

Um exemplo é a “função” alert.

```
alert('Olá, mundo!');
```

Essa é uma “função” do navegador, e um servidor não faz sentido termos um alert.

Mas o alert não é especificamente uma função, ela é uma método do objeto window, que é o objeto do browser.

```
window.alert('Olá, mundo!');
```

Construtores

Um construtor é utilizado para criar uma instância de um objeto.

Por exemplo, quando utilizamos o objeto date, passamos um operador `new`, esse operador é responsável por criar uma nova instância do objeto construtor date, exemplo:

```
var date = new Date();
```

Valores e referência

Quando atribuímos o valor de uma variável a outra o que é passado é o valor, conforme exemplo abaixo:

```
var x = 0;  
var y = x;
```

Já se com objetos o comportamento é um pouco diferente. Por exemplo:

```
var x = [1];  
  
var y = x;  
  
y.push(2);
```

No exemplo acima temos um array sendo atribuído ao x. O array no javascript é um tipo de objeto, quando passamos o valor de um objeto para uma variável, na verdade, não estamos passando o valor e sim uma referencia ao objeto original.

Sendo assim, nesse exemplo o y refere-se ao objeto de array presente no x, logo se o y sofrer alteração, essa alteração será aplicada para o objeto original também.

Função construtora

Uma função construtora é responsável por definir um modelo de métodos e propriedade que poderá ser instanciado, exemplo:

```
function Caneta(){  
  this.cor = 'azul';  
  this.preco = 3;  
  this.mudarCor = function(cor){  
    this.cor = cor;  
  }  
}
```

Quando criamos uma função construtora uma boa prática é deixar a primeira letra em maiúsculo.

Agora se seguindo o exemplo criamos uma variável e essa variável criar uma instancia dessa função ela herdará todas as propriedades e métodos.

```
var objetoCaneta = new Caneta();
```