

Objetos

Programação para Internet I
IFB - Tecnologia em Sistemas para Internet

Marx Gomes van der Linden

Objetos

- Em **Javascript**, objetos são estruturas de dados associativas
 - » Conjunto de pares **propriedade: valor**
- Também podem ser usados para programação orientada a objetos

Exemplo

```
let usuario = {  
  "nome completo": 'Alice Sousa Almeida',  
  "login": 'alice93',  
  "email": 'alice93@gmail.com',  
  'premium': true,  
  'pontos': 513  
};
```

Exemplo

- Quando a propriedade não tem espaços ou caracteres especiais, as aspas são opcionais:

```
let usuario = {  
  "nome completo": 'Alice Sousa Almeida',  
  login: 'alice93',  
  email: 'alice93@gmail.com',  
  premium: true,  
  pontos: 513  
};
```

Acesso a propriedades: método 1

- Utiliza-se a sintaxe `objeto["propriedade"]`
- Funciona em todos os casos

```
usuario["nome completo"] = 'Alice Sousa Barbosa';  
usuario['email'] = 'alice_novo@gmail.com';  
usuario['pontos']++;
```

Acesso a propriedades: método 2

- Utiliza-se a sintaxe `objeto.propriedade`
- Funciona quando a propriedade não tem espaços no nome

```
usuario["nome completo"] = 'Alice Sousa Barbosa';  
usuario.email = 'alice_novo@gmail.com';  
usuario.pontos++;
```

Acrescentando propriedades

- Novas propriedades podem ser adicionadas a um objeto existente

```
let cachorro = {  
  nome: 'Ted',  
  tamanho: 'médio'  
};  
  
cachorro.cor = 'marrom';  
  
console.log(cachorro);
```

Excluindo propriedades

- Para excluir uma propriedade, use o operador **delete**

```
let cachorro = {  
  nome: 'Ted',  
  tamanho: 'médio'  
};  
  
delete cachorro.tamanho;  
cachorro.cor = 'marrom';  
  
console.log(cachorro);
```


Adicionando novas propriedades

- O objeto também pode ser criado vazio

```
let x = {};  
  
x.nome = 'Outro objeto';  
x.descricao = 'Mais um teste';  
x.valido = false;  
  
console.log(x);
```

Loop for .. in

- Para percorrer todas as propriedades de um objeto, use:

```
for(let prop in objeto){  
    //...  
}
```

- A cada iteração:
 - » **prop** contém uma propriedade
 - » **objeto[prop]** contém o valor correspondente

Objetos como parâmetros

- Objetos são passados como parâmetros sempre como **referência**

```
let item = {  
  nome: 'Computador',  
  preco: 1500  
}  
  
function mudaPreco(obj){  
  obj.preco = 2000;  
}  
  
mudaPreco(item);  
console.log(item);
```

Métodos

- Um objeto pode conter funções
- São chamadas **métodos** e acessam o objeto por meio da palavra **this**

```
let dino = {  
  nome: 'Tyrannosaurus rex',  
  periodo: 'Cretáceo',  
  peso: 14000,  
  
  info: function() {  
    return `O ${this.nome} viveu no período `  
    + `${this.periodo} e pesava ${this.peso}kg.`;  
  }  
};
```

toString()

- O método toString() é chamado automaticamente quando um objeto é tratado como string

```
let dino = {  
  nome: 'Tyrannosaurus rex',  
  periodo: 'Cretáceo',  
  peso: 14000,  
  
  toString: function() {  
    return `O ${this.nome} viveu no período `  
    + `${this.periodo} e pesava ${this.peso}kg.`;  
  }  
};
```

Object.create

- É possível criar objetos a partir de um **protótipo** usando **Object.create**

```
let prot = {  
  nome: 'Novo usuário',  
  país: 'br',  
  tipo: 'comum'  
};  
  
let user1 = Object.create(prot);  
user1.nome = 'Cebolinha';  
  
let user2 = Object.create(prot);  
user2.nome = 'Donald Duck';  
user2.país = 'us';
```

Get e set

- As palavras-chave **get** e **set** podem ser usadas para criar métodos que são usados como propriedades
- O usuário do objeto não precisa saber se está chamando uma propriedade comum ou um método
 - » Útil para encapsulamento

Get e set

```
let quadrado = {  
  altura: 5,  
  
  get largura(){  
    return this.altura;  
  },  
  set largura(lg){  
    this.altura = lg;  
  },  
  
  get area(){  
    return this.altura ** 2;  
  },  
  set area(a){  
    this.altura = Math.sqrt(a);  
  }  
}
```


Objetos compostos

- Objetos podem conter outros objetos para formar uma **hierarquia de dados**

```
let familias = {  
  Flintstones: {  
    pai: 'Fred',  
    amigo: 'Barney'  
  },  
  Jetsons: {  
    pai: 'George',  
    mae: 'Jane',  
    filho: 'Elroy'  
  },  
  Simpsons: {  
    pai: 'Homer',  
    mae: 'Marge',  
    filho: 'Bart'  
  }  
};
```