

JSON

# JSON

- **JSON** (*JavaScript Object Notation*) é um modelo para armazenamento e transmissão de informações no formato texto. Apesar de muito simples, tem sido bastante utilizado por aplicações Web devido a sua capacidade de estruturar informações de uma forma bem mais compacta do que a conseguida pelo modelo XML, tornando mais rápido o *parsing* dessas informações.



# JSON

- A ideia utilizada pelo JSON para representar informações é simples: para cada valor representado, atribui-se um nome (ou rótulo) que descreve o seu significado. Essa sintaxe é derivada da forma utilizada pelo **JavaScript** para representar informações.

Exemplos do uso do nome (rótulo):

```
"dia": 25  
"peso": 75.800  
"url": "www.seusite.com.br"  
"valor": -10  
"trabalha": false
```

# JSON

- A partir dos tipos básicos, é possível construir tipos complexos como ***array***, onde os elementos são separados por vírgula “,” e entre colchetes “[ ]”.

```
[10, 8.5, 6, 9]
```

Ou

```
["Domingo", "Segunda", "Terça", "Quarta", "Quinta", "Sexta", "Sábado"]
```

# JSON

Ou

```
[  
  "Domingo",  
  "Segunda",  
  "Terça",  
  "Quarta",  
  "Quinta",  
  "Sexta",  
  "Sábado"  
]
```

# JSON

Ou com os tipos de dados misturados.

```
[  
  10,  
  "Janeiro",  
  False,  
  10.55,  
  NULL,  
  "Uma frase qualquer!"  
]
```

# JSON

Para formar uma matriz, veja o exemplo:

```
[  
    [7, 8, 9],  
    [4, 5, 6],  
    [1, 2, 3]  
]
```

Os dados também podem ser representados por objetos, neste caso utiliza-se as chaves “{}” como limitadores.

# JSON

- Os dados também podem ser representados por objetos, neste caso utiliza-se as chaves “{}” como limitadores.

```
{  
    "aula": "XML"  
    "ano":2018  
    "turma": A  
}
```



# JSON

Outro exemplo de objeto:

```
{  
  "Domingo" : "Folga",  
  "Segunda" : "Viagem",  
  "Terça"   : "Pagamentos",  
  "Quarta"  : "Piscina",  
  "Quinta"  : "Churrasco",  
  "Sexta"   : "Carteado",  
  "Sábado"  : "Festa"  
}
```

# JSON

No próximo exemplo, são dois objetos em um única estrutura. Os objetos são: *“admin”* e *“usuario”*.

```
{
  "admin":
  {
    "url": "https://www.google.com.br",
    "codigo": 10
  },
  "usuario":
  {
    "url": "https://www.bing.com/",
    "codigo": 10
  }
}
```

# JSON

Outra forma de realizar a representação de vários objetos do mesmo tipo, é no formato de *array*.

```
[  
  {  
    "aula": "XML"  
    "ano": 2018  
    "turma": A  
  },  
  {  
    "aula": "XML"  
    "ano": 2018  
    "turma": B  
  }  
]
```

# JSON

## *Falando Sobre: JSON x XML*

### *Semelhanças:*

- São modelos que representam informações no formato texto;
- Possuem natureza auto descritiva;
- São capazes de representar informação complexa;
- Podem ser utilizados para transportar informações em aplicações e são considerados padrões para representação de dados;
- São independentes de linguagem.

### *Diferenças:*

- JSON não é uma linguagem de marcação, não possui **tag** de abertura e fechamento;
- JSON representa as informações de forma mais compacta;
- JSON não permite a execução de instruções de processamento, algo possível em XML;
- JSON é tipicamente destinado para a troca de informações, enquanto XML possui mais aplicações.

# Prática em PHP

- Vamos ao primeiro exemplo de código em PHP. Nesse exemplo vamos realizar a leitura em dois objetivos, note que as informações foram atribuídas diretamente a uma variável do PHP, *\$user*.

```
<?php
$user = '
{
  "admin":
  {
    "url": "https://www.google.com.br",
    "codigo": 10
  },
  "usuario":
  {
    "url": "https://www.bing.com/",
    "codigo": 20
  }
}';

$dados = json_decode($user);

echo "<b>Admin:</b> ". $dados->admin->url . '<br />';
echo "<b>Código:</b> ". $dados->admin->codigo;
echo "<br><br>";
echo "<b>Usuário:</b> ". $dados->usuario->url . '<br />';
echo "<b>Código:</b> ". $dados->usuario->codigo;
?>
```

Documento: lendojson\_exemplo01.php

# Prática em PHP

Onde:

- `json_decode($user)`: Decodifica uma *string* JSON - (PHP 5 >= 5.2.0, PHP 7)

*Manual do PHP:*

[https://secure.php.net/manual/pt\\_BR/function.json-decode.php](https://secure.php.net/manual/pt_BR/function.json-decode.php)

Diferente do XML, precisamos decodificar o pacote de dados, para posteriormente realizar a sua leitura. O acesso a informação será feito através do *array \$dados*.

- *\$dados -> admin -> url*: Para acesso a informação a referência são os rótulos da estrutura.

# Prática em PHP

- No próximo exemplo, vamos fazer o acesso a vários objetos armazenados em um *array*,

```
<?php
$listadeprodutos = '
[
  {
    "titulo": "Chave de Fenda",
    "vlr": 2.50
  },
  {
    "titulo": "Alicate",
    "vlr": 4.70
  },
  {
    "titulo": "Martelo",
    "vlr": 10.50
  }
]';

$json = json_decode($listadeprodutos);

foreach($json as $registro){
    echo "<b>Produto</b> ". $registro->titulo . '<br />';
    echo "<b>Valor:</b> ". $registro->vlr;
    echo "<br><br>";
}
?>
```

Documento: lendojson\_exemplo02.php

# Prática em PHP

- O *array* de objetos em JSON está armazenado em ***\$listadeprodutos***, o próximo passo, assim como no exemplo anterior é decodificar o documento (***\$json*** = `json_decode($listadeprodutos)`), pronto!!! Usamos a instrução ***foreach()*** para percorrer todos os elementos e imprimir os dados (***\$registro->titulo*** e ***\$registro->vlr***).