

Estudo da linguagem JSON

Objetivo do json

1. JSON (JavaScript Object Notation) é um formato de intercâmbio de dados leve. É fácil para humanos ler e escrever. É fácil para as máquinas analisar e gerar. É baseado em um subconjunto do Padrão de Linguagem de Programação JavaScript ECMA-262 3ª Edição - dezembro de 1999 . JSON é um formato de texto totalmente independente da linguagem, mas usa convenções familiares aos programadores da família C de linguagens, incluindo C, C ++, C #, Java, JavaScript, Perl, Python e muitos outros. Essas propriedades tornam o JSON uma linguagem de intercâmbio de dados ideal. [Veja mais...](#)

Tipos de dados, sintaxe e exemplos

1. Os tipos de dados básicos do JSON são:
 1. **Number**: um número que pode ter sinal, uma parte fracionária separada por um ponto (. , como é usual em alguns países) e eventualmente usar a notação E exponencial, mas não pode incluir não-números como NaN. Não há distinção entre inteiros e números de ponto flutuante, refletindo o fato de que o JavaScript armazena qualquer número como ponto flutuante de dupla precisão, mas outras linguagens que implementem JSON podem ter diferenças na representação dos números.
 2. **String**: uma cadeia de zero ou mais caracteres Unicode. Strings são delimitados por aspas duplas (") e suportam a barra inversa (\) como caractere de escape. obs: Parecido com as string da linguagem c.
 3. **Boolean**: um dos valores true ou false, correspondendo aos valores lógicos verdadeiro e falso.
 4. **Array**: uma lista ordenada de zero ou mais valores, cada um podendo ser de qualquer tipo. Arrays são delimitados por colchetes ([]), dentro dos quais ficam os valores, também conhecidos como elementos, separados por vírgulas. O primeiro elemento é o de índice 0.
 5. **Object**: uma coleção não ordenada de pares atributo-valor onde os atributos (ou nomes ou chaves) são strings. Como os objects pretendem representar vetores associativos (ECMA-404), é recomendado, mas não obrigatório, que cada atributo seja único dentro de um objeto. Objects são delimitados por chaves ({ }) e usam vírgulas para separar cada par , enquanto que no par o atributo e o valor ficam separados por dois pontos (** : **).
 6. **null**: Valor vazio ou nulo (não confundir com o zero) representado pela palavra null.
2. **Espaços em branco** são permitidos com limites e ignorados em volta e entre elementos sintáticos, mas não dentro de um string. Só quatro caracteres são tratados assim: o **espaço**, a **tabulação horizontal(TAB)**, o **line feed(LF)** e o **carriage return(CR)**. Em particular, a marca de ordem de byte não deve aparecer em um documento JSON conforme o padrão. JSON não tem uma sintaxe

/

para comentários. Para troca de dados em um ecossistema aberto, JSON deve estar codificado em UTF-8.

3. Quando se trabalha com json é importante na sessão `<head></head>` a tag: `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`

Exemplo de um objeto e uma matriz em json

```
// Um objeto
{
  "Clientes": {
    "id": 1,
    "nome": "Paulo Sérgio da Silva Pacheco",
    "rua": "Rua Francisco oliveira de Souza",
    "numero": 15,
    "bairro": "Icarai",
    "municipio": "caucaia",
    "estado": "ceara"
  }
}

// Uma matriz
{
  "usuarios ": [
    {
      "nome": "primeiro nome",
      "site": "www.primeirousuario.com.br",
      "usuario_id": 1
    },
    {
      "nome": "segundo nome",
      "site": "www.segundousuario.com.br",
      "usuario_id": 1
    }
  ]
}
```

REFERÊNCIAS

1. [Introducing JSON](#)
2. [JSON wikipédia](#)

HISTÓRICO

- ✓ Criar este documento com nome do arquivo [json.md](#)
- ✓ Procurar site oficial da linguagem json e criar resumo e um link para o mesmo.
- ✓ Criar um exemplo de um objeto json e uma matriz json.
- ✓ Criar a sessão tipos de dados, sintaxe e exemplos.
- ✓ Criar a sessão referências e adicionar as referências na produção deste documento.
- ✓ Criar a página `json/index.html`
- ✓ Criar opção no menu `index.js` para acessar a página `json/index.html`