01

Armazenando muitos dados

Transcrição

Começando deste ponto? Você pode fazer o DOWNLOAD (https://github.com/alura-cursos/logica-de-programacao-I/archive/aula-9.zip) do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

O jogo que criamos consiste em adivinhar um número em um intervalo grande de valores. Para ajudar o usuário nessa tarefa, que tal se tivermos mais de um segredo? Por exemplo, podemos ter 6 números catalogados como segredo e, da mesma forma que antes, verificamos se o usuário acerta o chute ou não. Por exemplo, vamos inserir o seguinte em nosso código:

```
var segredo1 = 16;
var segredo2 = 34;
var segredo3 = 37;
var segredo4 = 42;
var segredo5 = 50;
var segredo6 = 58;
```

Mas, para que exista uma real comparação entre o número escolhido pelo usuário e aquele pensado pelo jogo, precisamos utilizar um `if` que verifique cada um dos números e faça o **ou** (com `||`). E no caso de querermos mais segredos? Nessa situação precisaríamos criar mais variáveis e mexer novamente no `if`.

Esse processo seria facilitado se houvesse uma maneira fácil de guardar várias variáveis, sem saber exatamente quantas. E há! Vamos criar uma variável `segredos` que guarda vários números:

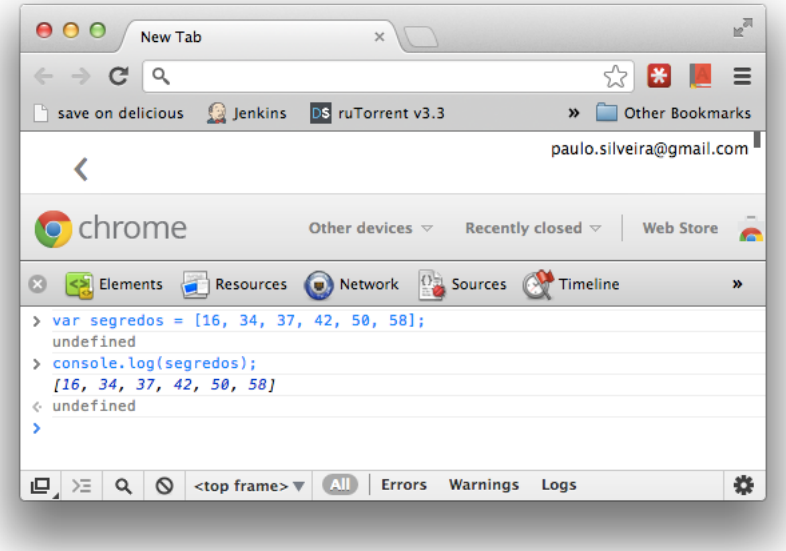
```
var segredos = [16, 34, 37, 42, 50, 58];
```

Essa sintaxe é nova para nós, nunca antes havíamos utilizado os colchetes. Neste caso eles criam uma **array**.

Abra o console do **Chrome** e faça alguns testes. O primeiro de todos é criar a variável `segredos` e imprimir seu valor.

```
var segredos = [16, 34, 37, 42, 50, 58];
console.log(segredos);
```

Repare na saída! Nada de surpreendente ocorre:



Como pegar o primeiro elemento, isto é, o primeiro número da série de segredos? Usamos os colchetes para indicar que queremos pegar um elemento e não a array inteira, observe:

```
console.log(segredos[1]);
```

Rodando isso o resultado é uma saída com valor 34! Por quê? O JavaScript, assim como outras linguagens conta as posições da array a partir do 0. Nesse caso. temos do número 0 ao 5 e para pegar o 16, faremos:

```
console.log(segredos[0]);
```

Se quisermos imprimir todos os números, basta fazer um `for` que vá de 0 a 5, como o observado abaixo:

```
for(var i = 0; i < 6; i = i + 1) {
    console.log(segredos[i]);
}
```

O `segredos[i]` acessa a array `segredos` na *i-ésima* posição! O número que vai dentro dos colchetes (`[]`) chamamos também de índice!

Uma das vantagens e facilidades das *arrays* é que podemos saber qual é o seu tamanho:

```
console.log(segredos.length);
```

Por esse motivo, é possível escrever o `for` sem fixar o número 6, substituindo-o por `segredos.length` , como:

```
for(var i = 0; i < segredos.length; i = i + 1) {
    console.log(segredos[i]);
}
```

Agora fica fácil! Basta criar o arquivo `loteria.html` e nele declaramos o campo de texto e o botão:

```
<input/>
<button>Compare com o meu segredo!</button>
```

Agora declaramos, no JavaScript, nossos segredos:

```
var segredos = [16, 34, 37, 42, 50, 58];
```

E, como já vimos, pegamos a caixa que na qual o usuário digita seu chute:

```
var input = document.querySelector("input");
```

Vamos preparar a função que nomearemos de *callback* e que será executada quando o evento de clique ocorrer. O que faremos dentro dela? Utilizaremos o `for` para percorrer todos os números e, assim, verificarmos se o "chute" equivale ao `caixaDoNumero.value` . Caso seja, imprimimos o que foi ganho e paramos! Sim, precisamos de um `break` para não continuar as verificações. Caso passe por todo o `for` , imprimimos a mensagem de que o usuário falhou!

```
function verifica() {

    var achou = false;

    for(var i = 0; i < segredos.length; i = i + 1) {
        if(segredos[i] == input.value) {
            achou = true;
            break;
        }
}

    if(achou == true) {
        alert("Parabéns! Você acertou um dos números secretos");
    }
    else {
        alert("Infelizmente você errou!");
    }
}
```

Agora, basta registrar se a função realmente equivale ao callback do evento de clique no botão, como também já aprendemos:

```
var button = document.querySelector("button");
button.onclick = verifica;
</script>
```

Abra a sua loteria e tente acertar um dos números.