FONTE

Strings — O básico

Strings são contidas entre aspas ‘simples’ ou “duplas”, caso um string começe com uma string simples e tenha uma dupla no meio da frase o seguinte resultado será imprimido. Ex: 'Texto normal, "texto entre aspas duplas" '.

Caracteres de escape na string

O exemplo acima também funciona se todas as strings forem duplas, porém com strings simples, para corrigir esse problema é necessário usar uma barra invertida / (caractere de escape), essa barra invertida faz com que o próximo caractere (normalmente uma aspa simples ou duplas) seja reconhecido como parte do texto.

Concatenando strings

Para concatenar strings é usado o sinal de +, ‘string1 ‘ + ‘string2’ retorna string1 string2, dá pra fazer isso com quantas strings quiser.

Números x strings

Não é possível concatenar umas string com um número, porque o navegador entende que o programa está tentando representar uma string como um número e isso não faz sentido, porém, representar um número como string, faz.
O objeto `Number` serve para converter um número string em um número normal.
Por outro lado, todo número tem um método chamado toString() que converterá para a string equivalente.

Métodos úteis de string

Descobrindo o comprimento de uma string

`length` serve para mostrar o comprimento da frase. Ex: nome da variável.length. Esse método é útil caso seja necessário mostrar uma sequência de nomes em ordem de comprimento, ou caso o usuário tenha que informar seu nome em um formulário, com length é possível informar o usuário se seu nome é maior que o comprimento do campo.

Recuperando um caractere de string específico

Para retornar qualquer caractere dentro de uma string é usando a **notação colchete** - isso é, adicionar colchetes ([]) no final do nome da variável. Dentro dos colchetes, incluir o número do caractere que deseja retornar.
Para recuperar o último caractere de *qualquer* string, dá para combinar essa técnica com a propriedade `length`. Ex: `browserType[browserType.length-1];`

Encontrando uma substring dentro de uma string e extraindo-a

Para saber se uma string menor está presente dentro de uma maior (*substring está presente dentro de uma string*). Isso pode ser feito usando o método `indexOf ()`, que leva um único parâmetro, a substring que desejada. Ex: nome da var.indexof('substring desejada');.
Caso a substring não seja encontrada o resultado retornado será -1.

Quando se sabe onde uma substring começa dentro de uma string e sabe em qual caractere quer que ela termine, `slice ()` pode ser usado para extrair isto. Ex: nome da var`.slice(0,3);.`
O primeiro parâmetro é a posição do caractere a partir da qual será iniciada a extração, e o segundo parâmetro é a posição seguinte do último caractere a ser extraído, porém o último caractere não é incluído.
Para extrair todos os caracteres de uma string a partir de um certo ponto é colocado somente um parâmetro no slice.

Mudando entre maiúsculas e minúsculas

Os métodos toLowerCase() para minúsculas e toUpperCase() para maiúsculas.

Atualizando partes de uma string

Pode-se substituir uma substring dentro de uma string com uma outra substring usando o método replace(). Ex: `nome var.replace('primeira parte','parte que substituirá');`.
Em um programa real, teria que setar o valor da variável para ser o resultado da operação; não apenas atualizar o valor da substring automaticamente. Assim seria preciso realmente escrever isso: `browserType = browserType.replace('moz','van');`.