... not significant.

As in Java, in-line // and multiline /\* ... \*/.

Unfortunately, the multiline delimiters can appear in string literals, making commenting out arbitrary blocks of codes unreliable.

There are no integers, or floating point of any size other than 64-bits.
There are no characters either.

xey is the same as x \* pow(10, e).

As in Python, single or double quote delimiters can be used.

16-bit fixed Unicode is used.

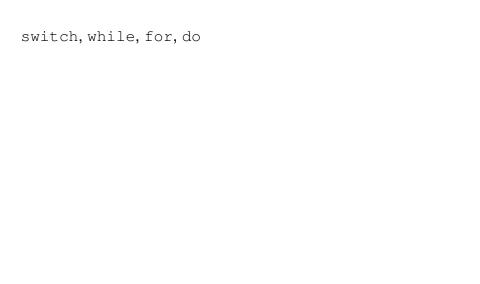
'string'.length

Use +.

The browser executes the compilation unit referred to in the tag right away.

They share a global namespace.

It indicates variables private to a function.



They don't define a new scope.

```
else {
To get a value:
if expr ? true-expr : false-expr
```

if (expr) then {

false, null, undefined, empty string, 0, NaN

```
do {
    ...
}
while (expr);
```

## Same as Java.

```
switch (expr) {
  case expr: ...
  default: ...
}
```

It falls through.

Identical to Java.

The first is as in C++/Java.

```
The second is a for-each: for (x in y) {
```

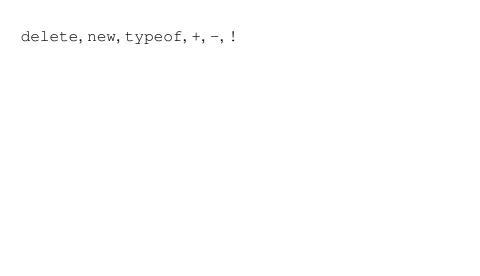
```
for (x in y) {
...
```

You can't know if the binding form was found in the iterable object's prototype chain. For that reason you often see the body of a for-each being a single condition of the form:

if (iterable.hasOwnProperty(bindingForm)) {

## The function's result is undefined.

=== and !==



'number', 'string', 'boolean', 'undefined',

'function', 'object'

```
function name(args) {
    ...
}
function(args) {
```

/regex/

It cam optionally be trailed by any of g, i, or m.

As is common, a comma-separated list inside square brackets.