

JS

Javascript

Fundamentals

var
function
if
for

Variables

```
var name = thing;
```

Variables

```
var name = 'Sam Tripp';
```

Variables

```
var name = 'Sam Tripp';
```

```
var age = 29;
```

```
var designer = true;
```

Variables

```
var name = 'Sam Tripp'; (String)
```

```
var age = 29; (Integer)
```

```
var designer = true; (Boolean)
```

```
age = 40;
```

```
age = false;
```

Variables

```
var age = 29;
```

```
if ( age > 18 ) { do something }
```

If statements

```
if ( something == something ) { }
```


If statements

```
var age = 29;
```

```
if ( age > 28 ) {
```

```
    alert( "You are old" );
```

```
}
```

Result

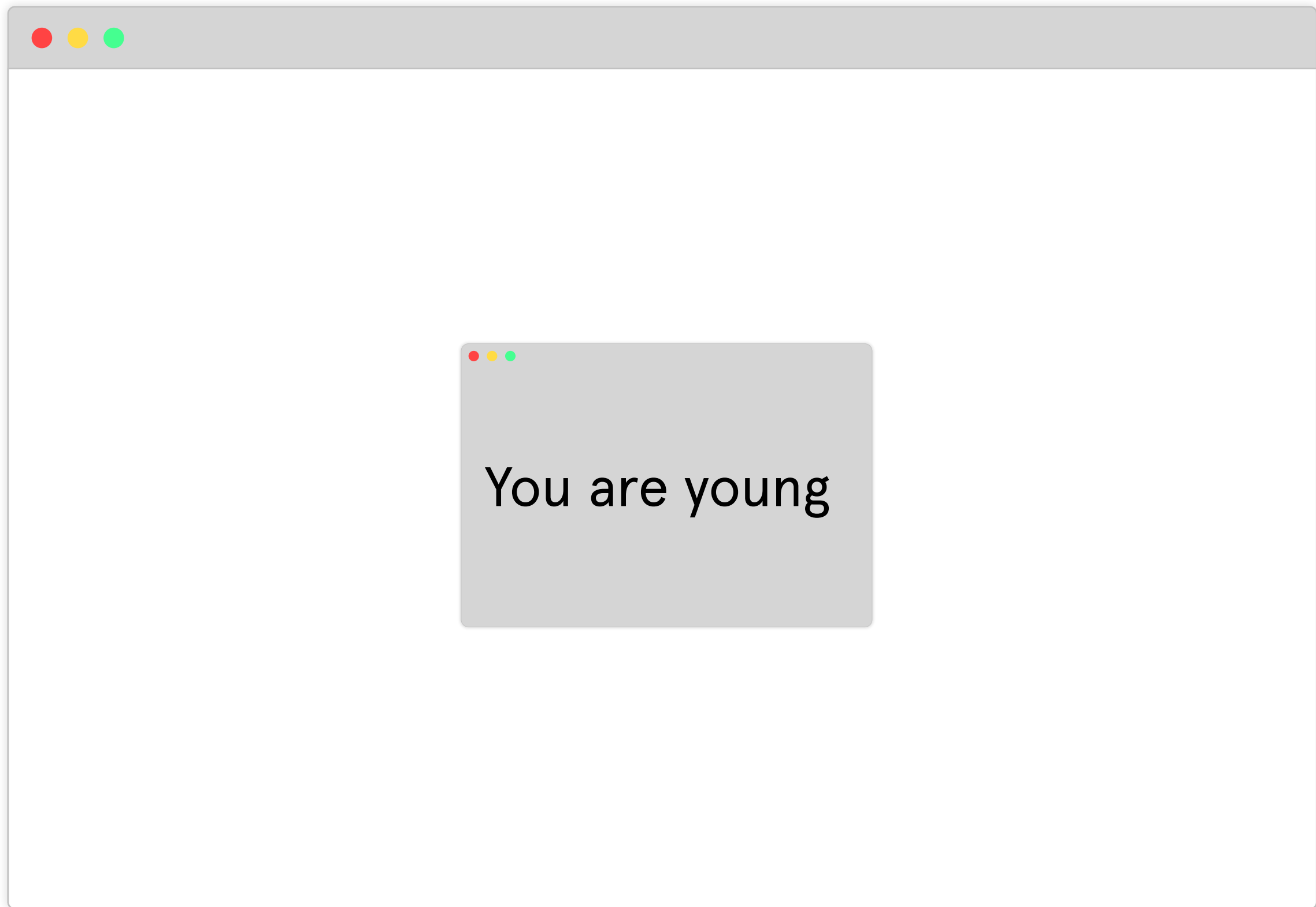


If statements

```
var age = 17;
```

```
if ( age > 18 ) {  
    alert( "You are old" );  
} else {  
    alert( "You are young" );  
}
```

Result



Functions

```
function name ( argument ) { }
```

Functions

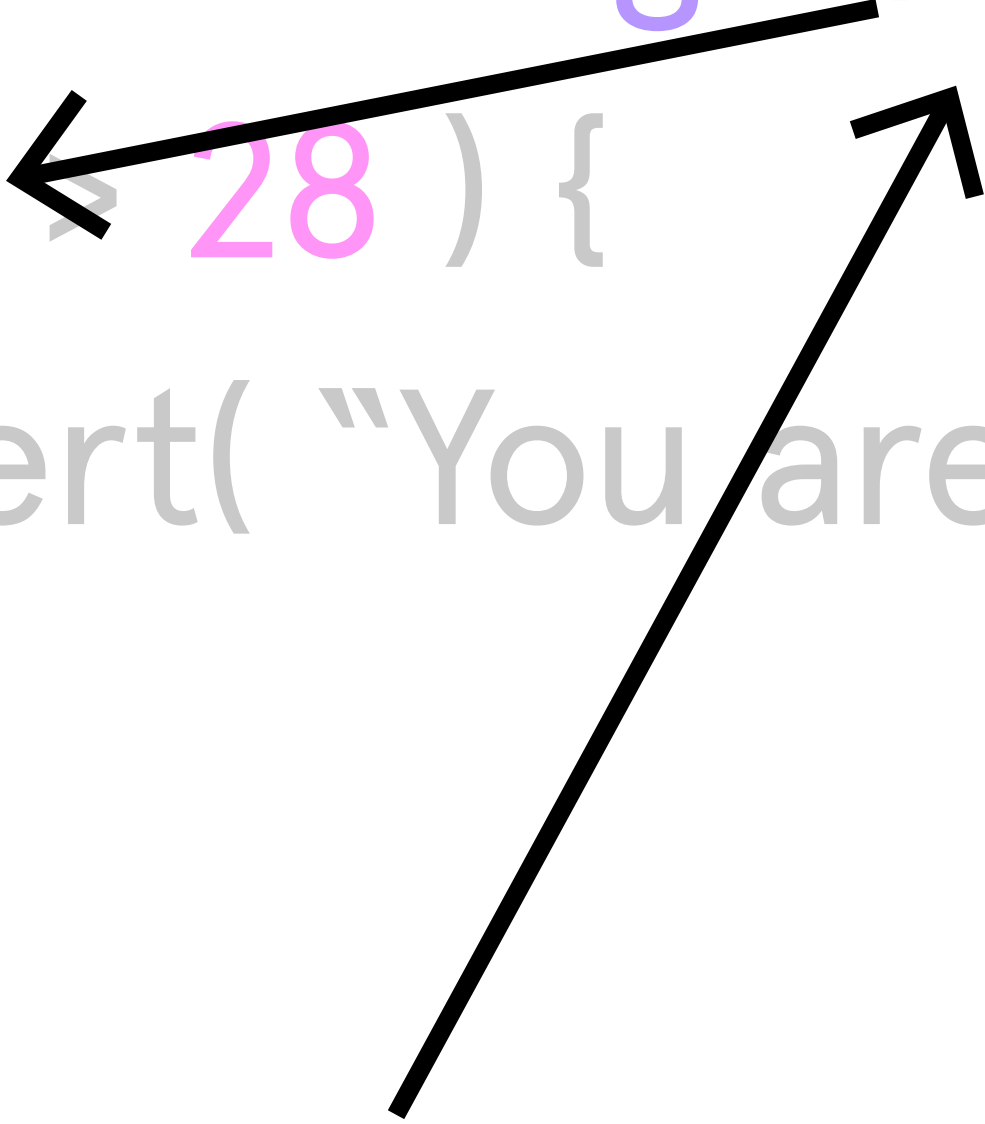
```
function checkAge ( a ) {  
    if ( a > 28 ) {  
        alert( "You are old" );  
    }  
}  
  
checkAge( 29 )
```

Functions

```
function checkAge ( a ) {  
    if ( a > 28 ) {  
        alert( "You are old" );  
    }  
}  
  
checkAge( 29 )  
checkAge( 32 )  
checkAge( 16 )
```

Functions

```
function checkAge ( a ) {  
  if ( a > 28 ) {  
    alert( "You are old" );  
  }  
}  
  
checkAge( 29 )
```



The diagram illustrates the flow of data in a function call. A black arrow originates from the argument '29' in the function call 'checkAge(29)' at the bottom and points to the parameter 'a' in the function definition 'function checkAge (a) {'. A second black arrow originates from the parameter 'a' in the function definition and points to the variable 'a' in the conditional statement 'if (a > 28) {', indicating how the argument is passed to the function and then used within its logic.

Functions

```
var age = 29;  
function checkAge ( a ) {  
    if ( a > 28 ) {  
        alert( "You are old" );  
    }  
}  
checkAge( age )
```

Functions

```
var age = 29;  
function checkAge ( a ) {  
    if ( a > 28 ) {  
        alert( "You are old" );  
    }  
}  
  
checkAge( age )  
age = 17  
checkAge( age )
```

For Loops

```
for ( num; condition; num+1 ) { }
```

For Loops

```
for ( var i = 0; i < 5; i++ ) {  
    do something;  
}
```


For Loops

```
var nums = [ 20, 18, 39, 3 ];  
for ( var i = 0; i < nums.length; i++ ) {  
    if ( nums[ i ] < 20 ) {  
        do something;  
    }  
}
```

For Loops

```
      0   1   2   3
var nums = [ 20, 18, 39, 3 ];

for ( var i = 0; i < nums.length; i++ ) {
    if ( nums[ i ] < 20 ) {
        do something;
    }
}
```



nums.length is 4. There are 4 things in the array

For Loops

```
      0   1   2   3
var nums = [ 20, 18, 39, 3 ];
for ( var i = 0; i < nums.length; i++ ) {
    if ( nums[ 0 ] < 20 ) {
        do something;
    }
}
```

The diagram illustrates the increment operation in a for loop. A straight arrow points from the initial value '0' in the loop condition to the first element of the array '20'. A curved arrow points from the 'i++' part of the loop condition to the '0' in the array access 'nums[0]', with the text 'Plus 1' next to it, indicating that the index is incremented by 1.

For Loops

	0	1	2	3
<code>var nums = [</code>	<code>20</code>	<code>18</code>	<code>39</code>	<code>3</code>
<code>];</code>				

```
for ( var i = 1; i < nums.length; i++ ) {  
    if ( nums[ 1 ] < 20 ) {  
        do something;  
    }  
}
```

Plus 1

For Loops

	0	1	2	3
var nums = [20	18	39	3];

```
for ( var i = 4; i < nums.length; i++ ) {
```

```
    if ( nums[ i ] < 20 ) {
```

```
        do something;
```

```
    }
```

```
}
```

i = 4

is 4 less than amount of nums?

No, it is the same amount.

Stop.