Boolean

A value that's either `true` or `false`.

`Boolean` values help to make decisions in code. They appear any time a logical condition is checked. For example, the condition "Is a mouse button being pressed?" must be either `true` or `false`:

```
// If the user presses the mouse, draw a circle at
// the mouse's location.
if (mouseIsPressed === true) {
  circle(mouseX, mouseY, 20);
}
```

The `if` statement checks whether mouseIsPressed is `true` and draws a circle if it is. `Boolean` expressions such as `mouseIsPressed === true` evaluate to one of the two possible `Boolean` values: `true` or `false`.

The `===` operator (EQUAL) checks whether two values are equal. If they are, the expression evaluates to `true`. Otherwise, it evaluates to `false`.

Note: There's also a `==` operator with two `=` instead of three. Don't use it.

The mouseIsPressed system variable is always `true` or `false`, so the code snippet above could also be written as follows:

```
if (mouseIsPressed) {
  circle(mouseX, mouseY, 20);
}
```

The `!==` operator (NOT EQUAL) checks whether two values are not equal, as in the following example:

```
if (2 + 2 !== 4) {
  text('War is peace.', 50, 50);
}
```

Starting from the left, the arithmetic expression `2 + 2` produces the value `4`. The `Boolean` expression `4 !== 4` evaluates to `false` because `4` is equal to itself. As a result, the `if` statement's body is skipped.

Note: There's also a `!=` operator with one `=` instead of two. Don't use it.

The `Boolean` operator `&&` (AND) checks whether two expressions are both `true`:

```
if (keyIsPressed === true && key === 'p') {
  text('You pressed the "p" key!', 50, 50);
}
```

If the user is pressing a key AND that key is `'p'`, then a message will display.

The `Boolean` operator `||` (OR) checks whether at least one of two expressions is `true`:

```
if (keyIsPressed === true || mouseIsPressed === true) {
  text('You did something!', 50, 50);
}
```

If the user presses a key, or presses a mouse button, or both, then a message will display.

The following truth table summarizes a few common scenarios with `&&` and `||`:

```
true && true  // true
true && false // false
false && false // false
true || true  // true
true || false // true
false || false // false
```

The relational operators `>`, `<< code="">, >= , and <=< code=""> also produce Boolean values:`

```
2 > 1 // true
2 < 1 // false
2 >= 2 // true
2 <= 2="" true="" <="" code=""/>
```

See if for more information about `if` statements and Number for more information about `Number`s.

Examples

```
function setup() {
  createCanvas(100, 100);

  describe('A gray square. When the user presses the mouse, a circle appears at that location.');
}

function draw() {
  background(200);

  // If the user presses the mouse, draw a circle at that location.
  if (mouseIsPressed) {
    circle(mouseX, mouseY, 20);
  }
}
```

```
function setup() {
  createCanvas(100, 100);

  describe('A gray square. When the user presses the mouse, a circle appears at that location.');
}

function draw() {
  background(200);

  // If the user presses the mouse, draw a circle at that location.
  if (mouseIsPressed === true) {
    circle(mouseX, mouseY, 20);
  }
}
```

```
// Click on the canvas to begin detecting key presses.

function setup() {
  createCanvas(100, 100);

  describe('A gray square that turns pink when the user presses the mouse or a key.');
}

function draw() {
  background(200);

  // If the user presses the mouse, change the background color.
  if (mouseIsPressed === true || keyIsPressed === true) {
    background('deeppink');
  }
}
```

```
// Click the canvas to begin detecting key presses.

// Create a Boolean variable.
let isPlaying = false;

function setup() {
  createCanvas(100, 100);

  describe(
    'The message "Begin?\nY or N" written in green on a black background. The message "Good luck!" appears when they press the "y" key.'
  );
}

function draw() {
  background(0);

  // Style the text.
  textAlign(CENTER, CENTER);
  textFont('Courier New');
  textSize(16);
  fill(0, 255, 0);

  // Display a different message when the user begins playing.
  if (isPlaying === false) {
    text('Begin?', 50, 40);
    text('Y or N', 50, 60);
  } else {
    text('Good luck!', 50, 50);
```

This page is generated from the comments in src/core/reference.js . Please feel free to edit it and submit a pull request!

Related References

class
A template for creating objects of a particular type.

console
Prints a message to the web browser's console.

for
A way to repeat a block of code when the number of iterations is known.

function
A named group of statements.