Object

A container for data that's stored as key-value pairs.

Objects help to organize related data of any type, including other objects. A value stored in an object can be accessed by name, called its key. Each key-value pair is called a "property." Objects are similar to dictionaries in Python and maps in Java and Ruby.

For example, an object could contain the location, size, and appearance of a dog:

```
// Declare the dog variable and assign it an object.
let dog = { x: 50, y: 50, size: 20, emoji: '🐶' };

// Style the text.
textAlign(CENTER, CENTER);
textSize(dog.size);

// Draw the dog.
text(dog.emoji, dog.x, dog.y);
```

The variable `dog` is assigned an object with four properties. Objects are declared with curly braces `{}`. Values can be accessed using the dot operator, as in `dog.size`. In the example above, the key `size` corresponds to the value `20`. Objects can also be empty to start:

```
// Declare a cat variable and assign it an empty object.
let cat = {};

// Add properties to the object.
cat.x = 50;
cat.y = 50;
cat.size = 20;
cat.emoji = '🐱';

// Style the text.
textAlign(CENTER, CENTER);
textSize(cat.size);

// Draw the cat.
text(cat.emoji, cat.x, cat.y);
```

An object's data can be updated while a sketch runs. For example, the `cat` could run away from the `dog` by updating its location:

```
// Run to the right.
cat.x += 5;
```

If needed, an object's values can be accessed using square brackets `[]` and strings instead of dot notation:

```
// Run to the right.
cat["x"] += 5;
```

This syntax can be helpful when the key's name has spaces, as in `cat['height (m)']`.

Examples

```
// Declare the variable data and assign it an object with three properties.
let data = { x: 50, y: 50, d: 20 };

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

  describe('A white circle on a gray background.');
}

function draw() {
  background(200);

  // Access the object's values using the . operator.
  circle(data.x, data.y, data.d);
}
```

```
// Declare the variable data and assign it an object with three properties.
let data = { x: 50, y: 50, d: 20 };

// Add another property for color.
data.color = 'deeppink';

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

  describe('A pink circle on a gray background.');
}

function draw() {
  background(200);

  // Access the object's values using the . operator.
  fill(data.color);
  circle(data.x, data.y, data.d);
}
```

```
// Declare the variable data and assign it an object with three properties.
let data = { x: 50, y: 50, d: 20 };

// Add another property for color.
data.color = 'deeppink';

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

  describe('A white circle on a gray background.');
}

function draw() {
  background(200);

  // Access the object's values using the . operator.
  fill(data.color);
  circle(data.x, data.y, data.d);

  // Update the object's x and y properties.
  data.x += random(-1, 1);
  data.y += random(-1, 1);
}
```

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.