Array

A list that keeps several pieces of data in order.

Arrays are helpful for storing related data. They can contain data of any type. For example, an array could contain a list of someone's favorite colors as strings. Arrays are created as follows:

```
let myArray = ['deeppink', 'darkorchid', 'magenta'];
```

Each piece of data in an array is called an element. Each element has an address, or index, within its array. The variable `myArray` refers to an array with three String elements, `'deeppink'`, `'darkorchid'`, and `'magenta'`. Arrays are zero-indexed, which means that `'deeppink'` is at index 0, `'darkorchid'` is at index 1, and `'magenta'` is at index 2. Array elements can be accessed using their indices as follows:

```
let zeroth = myArray[0]; // 'deeppink'
let first = myArray[1]; // 'darkorchid'
let second = myArray[2]; // 'magenta'
```

Elements can be added to the end of an array by calling the push() method as follows:

```
myArray.push('lavender');

let third = myArray[3]; // 'lavender'
```

See MDN for more information about arrays.

Examples

```
// Declare the variable xCoordinates and assign it an arra
// with three numeric elements.
let xCoordinates = [25, 50, 75];

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

  describe(
    'Three white circles drawn in a horizontal line on a gray background.'
  );
}

function draw() {
  background(200);

  // Access the element at index 0, which is 25.
  circle(xCoordinates[0], 50, 20);

  // Access the element at index 1, which is 50.
  circle(xCoordinates[1], 50, 20);

  // Access the element at index 2, which is 75.
  circle(xCoordinates[2], 50, 20);
}
```

```
// Declare the variable xCoordinates and assign it an arra
with three numeric elements.
let xCoordinates = [20, 40, 60];

// Add another element to the end of the array.
xCoordinates.push(80);

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

  describe('Four white circles drawn in a horizontal line on a gray background.');
}

function draw() {
  background(200);

  // Access the element at index 0, which is 20.
  circle(xCoordinates[0], 50, 20);

  // Access the element at index 1, which is 40.
  circle(xCoordinates[1], 50, 20);

  // Access the element at index 2, which is 60.
  circle(xCoordinates[2], 50, 20);

  // Access the element at index 3, which is 80.
  circle(xCoordinates[3], 50, 20);
}
```

```
// Declare the variable xCoordinates and assign it an empt
array.
let xCoordinates = [];

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

  // Add elements to the array using a loop.
  for (let x = 20; x < 100; x += 20) {
    xCoordinates.push(x);
  }

  describe('Four white circles drawn in a horizontal line on a gray background.');
}

function draw() {
  background(200);

  // Access the element at index i and use it to draw a circle.
  for (let i = 0; i < xCoordinates.length; i += 1) {
    circle(xCoordinates[i], 50, 20);
  }
}
```

```
// Declare the variable xCoordinates and assign it an empt
array.
let xCoordinates = [];

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

  // Add elements to the array using a loop.
  for (let x = 20; x < 100; x += 20) {
    xCoordinates.push(x);
  }

  describe('Four white circles drawn in a horizontal line on a gray background.');
}

function draw() {
  background(200);

  // Access each element of the array and use it to draw a
circle.
  for (let x of xCoordinates) {
    circle(x, 50, 20);
  }
}
```

```
// Declare the variable xCoordinates and assign it an empt
array.
let xCoordinates = [];

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

  // Add elements to the array using a loop.
  for (let x = 20; x < 100; x += 20) {
    xCoordinates.push(x);
  }

  describe(
    'Four white circles in a horizontal line on a gray background. The circles move randomly.'
  );
}

function draw() {
  background(200);

  for (let i = 0; i < xCoordinates.length; i += 1) {
    // Update the element at index i.
    xCoordinates[i] += random(-1, 1);

    // Use the element at index i to draw a circle.
    circle(xCoordinates[i], 50, 20);
  }
}
```

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.