Arrays

The array variable type

Array Operation Idiom

Chapter 11, Examples 11-1, 11-2, 11-3, 11-4, 11-5, 11-6, 11-7, 11-8

What is printed to the console?

A. undefined
B. 0
C. 2
D. 3
E. 5

let a = 2;

function setup() {
  something(a, 3);
  print(a);
}

function something(b, c) {
  b = c;
}
p5* circleracer (one racer)

    // the x-position of the racer
    let x = 0;

    function draw() {
        background(220);  // grey
        racerDraw(x, 40, 1);
        x += 1;
    }

    // draw a 30px diameter racer at position x,y
    // showing the racer number
    function racerDraw(x, y, number) {
        ...
    }

    Starter: https://editor.p5js.org/cs105/sketches/Z8ND6GN5o

p5* circleracers (three racers)

    // the x-position of the racers
    let x1 = 0;
    let x2 = 0;
    let x3 = 0;

    function draw() {
        ...

        racerDraw(x1, 40, 1);
        x1 += random(0, 3);

        racerDraw(x2, 80, 2);
        x2 += random(0, 3);

        racerDraw(x3, 120, 3);
        x3 += random(0, 3);
An array is a special kind of variable that holds multiple values. Each array has a name (just like a “simple variable”). Each value stored in the array is called an element. Each element is accessed using a unique index. The index is a whole number starting from zero: 0, 1, 2, …
An Array is Like an Apartment Postal Box

- The postal box is at an apartment address
  - (like the array name)
- Each cupboard in the postal box stores something
  - (like elements in an array)
- The postal box cupboards are uniquely numbered
  - (like the array index)

Declaring an Array

- “Simple” Variables (for comparison)
  ```javascript
  let x = 4;       // Number variable
  let z = true;    // Boolean variable
  ```

- Array Variables
  ```javascript
  let a = [];      // an empty array
  ```
Declaring an Array

- “Simple” Variables (for comparison)
  
  ```javascript
  let w = 4;       // Number variable
  let z = true;    // Boolean variable
  ```

- Array Variables
  
  ```javascript
  let a = [];       // an empty array
  let b = [17, 7, -10, 66, 5]; // array of 5 Numbers
  let sts = [true, false, false]; // array of 3 Booleans
  let pos = [true, 99, 5.5];  // array with 3 values
  ```

JavaScript arrays can hold different values

Assigning Values to Array Elements

- You assign values to each array element using the `index`
  
  ```javascript
  // declare array with 5 elements
  let b = [17, 7, -10, 66, 5];

  // assign a new value to index 3
  b[3] = 1000;
  ```

  element index
  
  b[3] is the 4th element
### Adding Elements to an Array

- You can **declare** an array with no elements:
  ```javascript
  // declare empty array
  let a = [];  // I’m an empty array.
  ```

- Then **assign** values to elements later:
  ```javascript
  // assign values to elements
  a[0] = 15;
  a[1] = 7;
  a[2] = -10;
  a[3] = 66;
  a[4] = 5;
  ```

### Using Arrays

- Use array elements just like variables

  ```javascript
  let b = [111, 222, 333, 1];  // declare array
  
  b[0] = 15;
  b[1] = random(-100, 100);
  print(b[2]);
  let c = b[0] * 100;
  ellipse(b[0], b[1], b[2], b[3]);
  if (b[3] > 5) { ...  // Now I have 5 elements!
  }  // I’m an empty array.
  ```
What is printed to the console?

```javascript
let bar = [];
function setup() {
  bar[0] = 5;
  bar[1] = 4;
  bar[2] = bar[0] - 1;
  bar[3] = 2;
  print(bar[2]);
}
```

What is printed to the console?
What is printed to the console?

```javascript
let bar = [];

function setup() {
  bar[0] = 5;
  bar[1] = 4;
  bar[2] = 3;
  bar[3] = 2;
  print(bar[4]);
}
```

circleracers (with array)

```
// the x-position of the racers
let x = [0, 0, 0];

function draw() {
  ...

  racerDraw(x[0], 40, 1);
  x[0] += random(0, 3);

  racerDraw(x[1], 80, 2);
  x[1] += random(0, 3);

  racerDraw(x[2], 120, 3);
  x[2] += random(0, 3);
}
```
Variables for Indices

- Use variables or return values for indices

  ```javascript
  let b = [];
  let i = 0;
  b[i] = 15;  // b[0] will be 15
  b[i + 1] = 7;  // b[1] will be 7

  b[random(0, 5)] = 99; // no. will not work
  b[floor(random(0, 5))] = 99; // yes!
  ```

  use `floor` to convert a decimal random number to a whole number

---

**p5** circleracers (with array)

// the x-position of the racers
let x = [0, 0, 0];

function draw() {
  ...
  let i = 0;

  racerDraw(x[i], 40, i + 1);
  x[i] += random(0, 3);
  i++;

  racerDraw(x[i], 80, i + 1);
  x[i] += random(0, 3);
  i++;

  racerDraw(x[i], 120, i + 1);
  x[i] += random(0, 3);
}

these 3 lines look like a loop iteration
```javascript
// the x-position of the racers
let x = [0, 0, 0];

function draw() {
    ...
    for (let i = 0; i < 3; i++) {
        // calculate the y position
        let y = 40 * (i + 1);
        racerDraw(x[i], y, 30, i + 1);
        // update the racer’s position
        x[i] += random(0, 3);
    }
}
```

### Array Length

- Arrays know their own length
  - Arrays are **objects**
  - you access array length using object **“dot syntax”**

```javascript
let b = [0, 0, 0, 0, 0];
print(b.length); // prints 5
print(b.length - 1]); // ok, prints 0
print(b[b.length]); // undefined!
```
circleracers (with array and loop)

let x = [0, 0, 0]; // array

function draw() {
    ...
    for (let i = 0; i < x.length; i++) {
        // calculate the y position
        let y = 40 * (i + 1);
        racerDraw(x[i], y, 30, i + 1);
        // update the racer’s position
        x[i] += random(0, 3);
    }
}

Initializing a Large Array

- Use a loop to initialize all array elements to the same starting value

let b = []; // declare array

// initialize 5 array elements to 0
for (let i = 0; i < 5; i++) {
    b[i] = 0;
}
circleracers (with array and loop)

let x = []; // array

function setup() {
    // initialize 6 racer starting positions
    for (let i = 0; i < 6; i++) {
        x[i] = 0;
    }
}

function draw() {
    ...
    for (let i = 0; i < x.length; i++) {
        // calculate the y position
        let y = 40 * (i + 1);
        racerDraw(x[i], y, 30, i + 1);
        // update the racer’s position
        x[i] += random(0, 3);
    }
}

https://editor.p5js.org/cs105/sketches/q9fHgYVuT

Array Index Values (and Potential Errors)

- the index must be between 0 and one less than array length
  - if array length is 5, the index can be 0, 1, 2, 3, or 4
  - otherwise, it may lead to a runtime error
    or it may be a logic error (unintentionally adding an element)

- Examples:
  let b = [15, 7, -10, 66, 5];
  print(b[-1]); // undefined
  print(b[5]); // undefined
  point(b[4], b[5]); // runtime error
  b[-1] = 123; // does nothing
  b[5] = 123; // adds a 6th element with value 123
let b = [15, 7, -10, 66, 5];

function setup() {
    print("start");
    print(b); // prints the whole array

    print("stop");
}

Tricks to Create a New Array with Specific Length

- To create an empty array of length N, assign a value to the element with index N - 1
  
  let a = [];
  a[99] = 0; // creates 100 elements
  print(a); // elements 0 to 98 are undefined
  print(a.length); // prints 100

- Can also set the length property of an array:
  
  let a = [];
  a.length = 100; // creates 100 elements
  print(a); // all elements are undefined
  print(a.length); // prints 100
How many Xs are printed to the console?

for (let i = 0; i < 5; i++) {
    print("X");
}

What is printed to the console?

let arr = [];
for (let i = 0; i < 5; i++) {
    arr[i] = i;
}
print(arr[4]);
Array Operation Idiom

- Looping through an entire array to apply some operation to elements is a common **programming idiom**.

```javascript
// initialize array elements to same value
for (let i = 0; i < arr.length; i++) {
    arr[i] = 50;
}

// initialize array elements to random value
for (let i = 0; i < arr.length; i++) {
    arr[i] = random(0, 100);
}
```

Array Operation Idiom: Draw all Elements

- Draw a simple bar graph to visualize the array

```javascript
// visualize each element as a thin vertical bar
for (let i = 0; i < arr.length; i++) {
    line(i, height, i, height - arr[i]);
}
```
Array Operation: Sum all Elements

- Compute the sum (i.e. total) of all elements in an array by adding them one by one to an *accumulator* variable.
- Example array:
  ```javascript
  arr = [5, 3, 10, 15, 7];
  ```
  ```javascript
  // compute the sum of all elements
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    sum += arr[i];
  }
  print("sum:", sum);
  ```

Array Operation: Average Element Value

- Sum all elements in an array and divide total by the length of the array to compute the average element size.
- Example:
  ```javascript
  // calculate the average value
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    sum += arr[i];
  }
  let average = sum / arr.length;
  print("average:", average);
  ```
Array Operation: Find Largest Element Value

- Search the array to find the largest element value
  - Start by guessing that the largest value is the first element
  - check all other elements one-by-one to see if any are larger
  - if larger value is found, it becomes new largest value found so far

```javascript
// find the largest element value
// (assuming arr has at least length 1)
let largest = arr[0];
for (let i = 1; i < arr.length; i++) {
  if (arr[i] > largest) {
    largest = arr[i];
  }
}
print("largest:", largest);
```

Array Operation: Find Largest Element Index

- Search the array to find the index with largest element value
  - Start by guessing that index 0 has element with largest value
  - check all other elements one-by-one to see if any are larger
  - if larger one is found, use it’s index as the largest element index

```javascript
// find the index of the largest element
// (assuming arr has at least 1 element)
let indexOfLargest = 0;
for (let i = 1; i < arr.length; i++) {
  if (arr[i] > arr[indexOfLargest]) {
    indexOfLargest = i;
  }
}
print("index of largest:", indexOfLargest);
print("largest:", arr[indexOfLargest]);
```
Using Arrays with User-Defined Functions

Functions can have arrays as a parameter
Functions can return an array

What is printed to the console?

A. 1
B. 2
C. 3
D. 4
E. undefined
Functions Can Use an Array as a Parameter

// returns largest element in array a
// (assuming a has at least 1 element)
function largestValue(a) {
  let largest = a[0];
  for (let i = 1; i < a.length; i++) {
    if (a[i] > largest) {
      largest = a[i];
    }
  }
  return largest;
}

Functions Can Return an Array

// returns an array of size n with all
// elements set to value
function createArray(n, value) {
  let a = [];
  for (let i = 0; i < n; i++) {
    a[i] = value;
  }
  return a;
}

- Usage
  // create length 100 array with all elements 0
  let arr = createArray(100, 0);

- Variation: createRandomArray with all element values initialized to
  random number in between low to high

This kind of function is called a “factory” since it returns a new array.
arrayoperations

Demo code with all array operations shown on previous slides

https://editor.p5js.org/cs105/sketches/37V9fQ8Fe

forest

let treeX = [166, 90, 77, 30, 110, 131, 55];
let treeLeafColour = [];

function setup() {
...
    // initialize to random tint of green
    for (let i = 0; i < treeX.length; i++) {
        treeLeafColour[i] = random(30, 130);
    }
}

function draw() {
...
    // draw trees for forest
    for (let i = 0; i < treeX.length; i++) { 
        tree(treeX[i], 95, treeLeafColour[i]);
    }
}

https://editor.p5js.org/cs105/sketches/_WSaaW879
What is printed to the console?

```javascript
let a = [1, 2, 3, 4];
let v = 1;
for (let i = 0; i < a.length; i++) {
v = v * a[i];
}
print(v);
```

---

**snake**

```javascript
// Shift array elements
for (let i = 0; i < x.length - 1; i++) {
  // Shift all elements left one spot
  // x[0] = x[1], x[1] = x[2], and so on.
  // Stop at the second to last element
  x[i] = x[i + 1];
y[i] = y[i + 1];
}

// save the mouse position in the last element
x[x.length - 1] = mouseX;
y[y.length - 1] = mouseY;
```

---

https://editor.p5js.org/cs105/sketches/iUJs0a07u
https://editor.p5js.org/cs105/sketches/CF4-V1DAJ
### linegraph

```javascript
let m = 10; // margin

// draw a line graph
for (let i = 0; i < a.length; i++) {
  // draw the points
  let x = map(i, 0, a.length - 1, m, width - m);
  let y = map(a[i], 0, 100, height - m, m);
  strokeWeight(6);
  point(x, y);
  // draw the connecting lines
  if (i > 0) {
    let px = map(i - 1, 0, a.length - 1, m, width - m);
    let py = map(a[i - 1], 0, 100, height - m, m);
    strokeWeight(1);
    line(px, py, x, y);
  }
}

https://editor.p5js.org/cs105/sketches/h7JsYsv2z

https://editor.p5js.org/cs105/sketches/5Cxi3Yg1
```

### Sorting an Array

```javascript
let a = [3, 1, 2];

// sort elements in ascending order
a = sort(a); // a will be [1, 2, 3]
```
Sorting out Sorting
- https://youtu.be/HnQMDkUFzh4

Selection Sort - 35 comparisons, 67 array accesses, 0.50 ms delay
http://panthema.net/2013/sound-of-sorting

15 Sorting Algorithms in 6 Minutes
- https://youtu.be/kPRA0W1kECg
isArraySorted

// returns true if array a is sorted
// in ascending order, otherwise returns false
function isArraySorted(a) {
...

Test Examples:
isArraySorted([2, 3, 1, 6, 10]) returns false
isArraySorted([2, 3, 6, 6, 10]) returns true
isArraySorted([2]) returns true
isArraySorted([]) returns true

https://editor.p5js.org/cs105/sketches/17jeCGLfb

What is printed to the console?

A. 0
B. 2
C. 5
D. 10
E. 15

let a = [2, 3, 4, 1, 5];
let v = 0;
for (let i = 0; i < a.length; i++) {
v = v + a[i];
}
print(v);
Asymptotic Runtime of Algorithms

- O(1) and O(N) ... “big Oh”
- How many steps to assign and print one element?
  
  ```javascript
  arr[10] = 999;
  print(arr[10]);
  ```

  Does the runtime depend on the size of the array?

- How many steps to computer sum of all elements?
  ```javascript
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    sum += arr[i];
  }
  print("sum:", sum);
  ```

  Does the runtime depend on the size of the array?

We’ll consider a step to be each line of code executed.

How long does code take to run?

```javascript
// save current time (in milliseconds) to t
let t = millis();

// <code you want to time goes here>

// see how time has elapsed since t was created
print(millis() - t);
```

millis() returns the number of milliseconds since the program started.

https://editor.p5js.org/cs105/sketches/37V9fQ8Fe
Does this runtime depend on the size of the array?