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
circleracer (one racer)

Starter: https://editor.p5js.org/sanghosuh/sketches/6KWnVHN9
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/sanghosuh/sketches/6KWnVHN9
https://editor.p5js.org/cs105/sketches/Z8ND6GN5o
circleracers (three racers)

// create three racers
// each racer moving at different speeds
// in each frame

Starter: https://editor.p5js.org/sanghosuh/sketches/1mAocGv
```javascript
// 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);
}
```

https://editor.p5js.org/sanghosuh/sketches/dqCa_Zc7
circleracers (20 racers)

```javascript
let x1 = 0;
let x2 = 0;
let x3 = 0;
let x4 = 0;
let x5 = 0;
let x6 = 0;
let x7 = 0;
let x8 = 0;
let x9 = 0;
let x10 = 0;
let x11 = 0;
let x12 = 0;
let x13 = 0;
let x14 = 0;
let x15 = 0;
let x16 = 0;
let x17 = 0;
let x18 = 0;
let x19 = 0;
let x20 = 0;

racerDraw(x1, 40, i);
    x1 += random(0, 3);
    racerDraw(x11, 440, i);
    x11 += random(0, 3);

racerDraw(x2, 80, i);
    x2 += random(0, 3);
    racerDraw(x12, 480, i);
    x12 += random(0, 3);

racerDraw(x3, 120, i);
    x3 += random(0, 3);
    racerDraw(x13, 520, i);
    x13 += random(0, 3);

racerDraw(x4, 160, i);
    x4 += random(0, 3);
    racerDraw(x14, 560, i);
    x14 += random(0, 3);

racerDraw(x5, 200, i);
    x5 += random(0, 3);
    racerDraw(x15, 600, i);
    x15 += random(0, 3);

racerDraw(x6, 240, i);
    x6 += random(0, 3);
    racerDraw(x16, 640, i);
    x16 += random(0, 3);

racerDraw(x7, 280, i);
    x7 += random(0, 3);
    racerDraw(x17, 680, i);
    x17 += random(0, 3);

racerDraw(x8, 320, i);
    x8 += random(0, 3);
    racerDraw(x18, 720, i);
    x18 += random(0, 3);

racerDraw(x9, 360, i);
    x9 += random(0, 3);
    racerDraw(x19, 760, i);
    x19 += random(0, 3);

racerDraw(x10, 400, i);
    x10 += random(0, 3);
    racerDraw(x20, 800, i);
    x20 += random(0, 3);
```
Inefficient
Array

- An array is a special kind of variable that holds multiple values.

“Simple” Variable

Array Variable

a

b[]

23

0 10 0 -10 35
Array

- Each array has a **name** (just like a “simple variable”)

\[
\begin{array}{c}
\text{b} [] \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c}
0 & 10 & 0 & -10 & 35 \\
\end{array}
\]
Array

- Each *value* stored in the array is called an *element*. 

| 0 | 10 | 0 | -10 | 35 |
- Each *element* is accessed using a unique **index**
  - The index is a *whole number* starting from zero: 0, 1, 2, ...

```
  0    10    0   -10    35
```
Common Confusion / Misconception: Index

- Why is the first element in the array $b[0]$ and not $b[1]$?
Common Confusion / Misconception: Index

- **Wrong** way to think of **index**:
  - as an *ordinal number*

  e.g. 1 in \texttt{b[1]} means the “first” element of the list
  2 in \texttt{b[2]} means the “second” element of the list

**Ordinal Number is...**

“An ordinal number is a number that tells the *position* of something in a list, such as *1st, 2nd, 3rd, 4th, 5th* etc. Most ordinal numbers end in "th" except for: one ⇒ first (1st) two ⇒ second (2nd) three ⇒ third (3rd)”

Credit: [https://www.mathsisfun.com/numbers/cardinal-ordinal-chart.html](https://www.mathsisfun.com/numbers/cardinal-ordinal-chart.html)
Common Confusion / Misconception: Index

- **Wrong** way to think of index:
  - as an *ordinal number*

  e.g. 1 in $b[1]$ means the *first* element of the list

  2 in $b[2]$ means the “second” element of the list

Credit: [https://www.mathsisfun.com/numbers/cardinal-ordinal-chart.html](https://www.mathsisfun.com/numbers/cardinal-ordinal-chart.html)
Common Confusion / Misconception: Index

- **Right** way to think of **index**:
  - as an *offset (i.e. distance)*

  e.g. 0 in \(b[0]\) means 0 element away from \(b\)
  1 in \(b[1]\) means 1 element away from \(b\)
**location**  lilypad
location: lilypad + 1
<table>
<thead>
<tr>
<th>location</th>
<th>lilypad</th>
<th>lilypad + 1</th>
<th>lilypad + 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+1</td>
<td>+1</td>
<td></td>
</tr>
</tbody>
</table>
location  lilypad[0]
+1

| location | lilypad[0] | lilypad[1] |
|----------|------------|------------|------------|

+1 +1
index represents offset,
NOT position
Another Analogy: 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)
  ```
  let x; // Number variable
  let z; // Boolean variable
  ```

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

array name

square brackets for the list of elements
Initializing an Array

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

- Array Variables
  ```javascript
  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**

  // 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:
  // declare empty array
  let a = [];

• Then assign values to elements later:
  // 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]; // initialize 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) { ...
```
```
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);
```
// 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);

Convert this code to use array
circleracers (with array)

Starter: https://editor.p5js.org/sanghosuh/sketches/w6PHfjxg
 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);
}

https://editor.p5js.org/sanghosuh.sketches/CTXSjO1c
Assigning indices using hard-coded numbers

- We hard-coded values for indices

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

“Hard coding is a term used in computer programming. It means using specific amounts or value instead of using calculated or referenced amounts.”

Credit: https://www.techwalla.com/articles/what-is-hard-coding-in-excel
Assigning indices using variables

- You can 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` (or ceil) to convert a decimal random number to a whole number
```javascript
// 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);
}
```
// 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);
}

Convert this code to use variables as indices
circleracers (with array)

Starter: https://editor.p5js.org/sanghosuh/sketches/eM_bqYc2
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);
}

Starter: https://editor.p5js.org/sanghosuh/sketches/eM_bqYc2

these 3 lines look like a loop iteration
```javascript
// 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);
}
```

Starter: [https://editor.p5js.org/sanghosuh/sketches/eM_bqYc2](https://editor.p5js.org/sanghosuh/sketches/eM_bqYc2)
circleracers (with array and loop)

Starter: https://editor.p5js.org/sanghosuh/sketches/a8wSffF3
// 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);
  }
}

https://editor.p5js.org/sanghosuh/sketches/e1fRIITLB
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!

say “b dot length”
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);
  }
}
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);
  }
}

Q) What if we have 20 racers?
Do we need to write zero 20 times?
Is there a better way to do this?
Initializing a Large Array

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

```javascript
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)

function setup() {
  // initialize 6 racer starting positions
  ...
}

draw() {
  // draw racers
  ...
}
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);
  }
  
"array “initialization”"
Array Index Values (and Potential Errors)

- the index must be between 0 and one less than array length
  - e.g. 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)
Array Index Values (and Potential Errors)

- 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");
}

https://editor.p5js.org/cs105/sketches/Mmi1KZ5lw
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 $N - 1$
  
  ```javascript
  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:
  
  ```javascript
  let a = [];
  a.length = 100; // creates 100 elements
  print(a); // all elements are undefined
  print(a.length); // prints 100
  ```
"What will Hero do?"

- Fight
- Run
- Get Help
Modular Design - A6

- Modules
  - controller
  - character
  - story
  - background

“What will Hero do?”

- Fight
- Run
- Get Help