

### Parameterized Shapes



### Color

- A data type that represents an RGB color  
`color oliveGreen = color(85, 107, 47);`
  - Functions that return a color component
    - `red()`                    `color c = color(20, 20, 140);`
    - `green()`
    - `blue()`
- ```

float r = red(c);
float g = green(c);
float b = blue(c);

fill(r, g, b);
fill(color(r, g, b));
fill(c);

```

### Example

- gradient

### So far..

- A program consists of actions:
  - call drawing functions
    - line, rect, ellipse, etc.
  - change the drawing state
    - size, background, stroke
  - compute
    - \*,+,-,/,%,cos, etc.
  - input
    - mouse
    - keyboard
- Actions happen sequentially unless
  - literals
    - 1,2,3,'a',"hello",1.0,true, etc.
  - variables
    - int x;
    - boolean test;
    - etc.
  - Actions happen sequentially unless
    - `if(condition){}else if(condition){}else{}`
    - `while(){}, for(){}`
    - `functionCall();`
- Actions are done on:

### Variables

- So far
  - store values for re-use
  - single value
  - declare before use
- What if you need many many variables?
- New concept
  - store a group of values
- A sequence or collection of values
  - {1,2,3,4}
  - {2,4,6,8}
  - {1,3,5,7}
  - {1,2,3,1,2,1,1,1,1,5,4,3,5,0,2,4,3,1,6,3,7,2,3,2,2,7,7,7,6,5,4,4}

### Array, Variable Grouping

- a fixed size
- one type of value
- declare an array
  - `int[] intervals;`
  - `float[] temps;`
- instantiate an array
  - `intervals = new int[10];`
  - `temps = {1.0,32.0,212.0};`
- assign values to elements of an array
  - `intervals[0] = 10;`
  - `temps[2] = -300.0;`

### Arrays

- A special kind of variable that holds not one, but many data items of a given type.
- Declared like variables, only type is followed by a pair of brackets.

```
float[] xs;
```

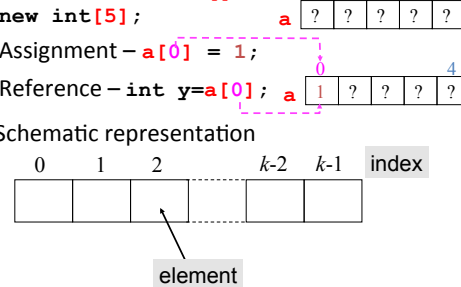
- Can be instantiated using a special syntax involving the **new** keyword, the type, and a size in brackets.
- Or initialized with a list of values

```
int[] primes = new int[10]; // Ten primes
```

```
int[] primes = {2, 3, 5, 7};
```

### Arrays

- Declaration – `int a[] = new int[5];`
- Assignment – `a[0] = 1;`
- Reference – `int y=a[0];`
- Schematic representation



### Arrays

- Individual data items are accessed with an index and square brackets.
  - `primes[0], primes[1], etc`
  - **Indexes start at 0!**
- The length of an array can be determined using its `length` property.
  - `primes.length`
  - The length of an array is one greater than the last valid index.
- Arrays can be passed to, and returned from functions.

```
int[] diameters = new int[10];

void setup() {
  size(500, 500);

  // fill with random values between 0 and width/2
  for (int i=0; i<diameters.length; i++) {
    diameters[i] = int(random(width/2));
  }

  fill(255, 0, 0);
  for (int i=0; i<diameters.length; i++) {
    ellipse(random(width), random(height), diameters[i],
            diameters[i]);
  }
}
```

### Array as parameter of a function

```
void drawCircles(int diameter[]) {
  for (int i=0; i < diameter.length; i++) {
    // draw the circle
    ellipse(random(width), random(height),
            diameter[i], diameter[i]);
  }
}
```