Fractional values of numbers with `double`

Use a double to store a number that has values after the decimal point.

The `int` data type only represents numbers without a fractional part. What if we want to store an approximation for π in a mathematics program? We can use a different data type, called a `double`, to store numbers that have values after the decimal point. `double` is short for *double-precision floating point*. Floating point means that there is a decimal point that can be placed at different locations (or float), in the number.

Computer programming languages evolve over time; double-precision just means that this representation allows double the precision that an earlier representation of floating-point numbers used. There is also a floating point type called `float` in Java, which uses less memory and has less precision than a double. It is rarely used.

Exercise: circle area

Objective: Create and use values and variables of the `double` type.

Declare and give initial values to variables `r` and `pi` representing the radius of a circle, and an approximation of the mathematical constant π. Use those values to compute and print the area of the circle. You can square a number in Java by multiplying it by itself.

CircleArea.java | Sample Solution

```
class CircleArea {
  public static void main(String args[]) {
        double area;
  }
}
```

Notice from the sample solution that you can declare multiple variables of the same type using a single line of code.

Exercise: big smile

Objective: use variables to allow the behavior of code to be changed easily.

The code below will draw a smiley face on the screen, and you can use the variables `x` and `y` to change where the smiley face is drawn. But the smiley face is always the same size. Add a new variable, `scale`, that allows you to change the size of the smiley face to make the face either larger or smaller. For example, if `scale` had the value `2`, then the code would draw the smiley face twice as large (but still centered on `x` and `y`).

Any time you create a variable, you should first ask what type of data it will hold, since you will need that information in order to declare the variable. Should `scale` be an integer? No, not if you'd like it to hold a value like `0.5` to make the smiley smaller.

Test your code by changing variable values a few times to draw the smiley face at different locations at both small and large scales.

Hint: Some values should be scaled; others should not be. For example, if the *x* location of the left eye is `x - 20`, then the `20` should be scaled, but the `x` should not be, since `x` will still be the x location of the center of the entire smiley, even if the smiley is scaled.

BigSmile.java   Sample solution

```
import com.educative.graphics.*;

class BigSmile {
  public static void main(String[] args) {
        Canvas c;
    c = new Canvas(200, 200);

    int x;
    int y;
    x = 100;
    y = 100;

    // Draw the outline of the face
    c.fill("yellow");
    c.stroke("black");
    c.circle(x, y, 50);

    // draw the mouth
    c.stroke("black");
    c.fill("yellow");
    c.circle(x, y, 30);
    c.stroke("yellow");
    c.rect(x - 32, y - 32, 62, 40);

    // draw the eyes
    c.stroke("black");
    c.fill("black");
    c.circle(x - 20, y - 10, 5);
    c.circle(x + 20, y - 10, 5);

  }
}
```