Variables

Variables are pieces of a computer's memory used to store values. All variables have a type, a name and a value. Types determine what kind of data can be stored in a particular variable.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>whole numbers</td>
<td>2, 4, 6, 8</td>
</tr>
<tr>
<td>float</td>
<td>real numbers (decimals)</td>
<td>1.2, 3.1415, 76, 12.1</td>
</tr>
<tr>
<td>String</td>
<td>text</td>
<td>&quot;Jackie likes puppies&quot;, &quot;Maya&quot;</td>
</tr>
</tbody>
</table>

(type) (name) = (value);

Creates a variable with the given name and type and stores a value into it.

    int age = 14;
    float width = 7.4;
    String name = "Jenny";

(name) = (value);

Changes the value of an existing variable with the given name.

    age = 15;
    age = age + 1;
    name = "Sally";

Processing Built-In Variables

Don't need to be declared or initialized. Use them anywhere real numbers are used.

- mouseX - current x position of the mouse cursor
- mouseY - current y position of the mouse cursor
- width - width of the screen
- height - height of the screen

Animation in Processing

To create an animation, all you need to do is to quickly quickly draw different images one after the other! Images are often changed by variables.

```java
void setup() {
    // any code written here will be run exactly once at the beginning
    size(400, 400);  // size of the window
    smooth();
}

void draw() {
    // any code written here will be run 60 times a second
    background(255);  // clear background for no trail
}```
Variables and Animation Challenges

1. Write a program to draw an 'X' across the screen that resizes based on the window's size.

2. The following code draws a gray square in the upper left corner of the screen. Modify it so that the square starts by filling the screen and shrinks over time until it disappears.

```cpp
void setup() {
  size(200, 200);
  background(255);
  smooth();
}

void draw() {
  background(255);
  fill(200);
  rect(0, 0, 10, 10);
}
```

3. What changes about the output if the `background(255)` line is taken out of `draw()` (try it if you're not sure)?

4. What determines the speed at which the square shrinks?

5. Grab the code from the website and experiment with it. Can you make the shrinking speed change?