Review

- Dropbox
- Processing folder structure
- Drawing Images
- Variables
- Variable types
- Integer division
- Conditionals: if - else if - else
- Motion simulation

Expressions

- Collections of data values and variables related by operators and function calls, and grouped by parentheses.
- Expressions are automatically evaluated and replaced by the final evaluated value.
- Expressions can be assigned to variables using "="
  - Expression is always on right
  - Variable name is always on left

```
variable_name = expression;
```

Some Built-in Mathematical Functions

- \(\sin(x)\), \(\cos(x)\), \(\tan(x)\), \(\text{asin}(x)\), …
- \(\text{abs}(x)\), \(\exp(x)\), \(\text{pow}(x, y)\), \(\log(x)\), \(\text{sqrt}(x)\), …
- \(\text{max}(x_1, x_2)\), \(\text{min}(x_1, x_2)\), \(\text{floor}(x)\), \(\text{ceil}(x)\), …

\(\text{dist}(x_1, y_1, x_2, y_2)\) \(\rightarrow\) distance between two points
\(\text{norm}(\text{value, low, high})\) \(\rightarrow\) normalizes a value to [0-1]

... and many more, all of which can be included in an expression.

Evaluating Expressions

```
1 + 2
\text{pow}(\sin(x),2) + \text{pow}(\cos(x),2) \approx 1.0
\text{max}(1, 2, 3) ≥ 2
\text{floor}(2.9) = \text{ceil}(1.8)
```

Iteration

Repetition of a program block

- Iterate when a block of code is to repeated multiple times.

Options

- The while-loop
- The for-loop

Iteration: while-loop

```
while (boolean_expression) {
    statements;
    // continue;
    // break;
}
```

- Statements are repeatedly executed while the boolean expression remains true;
- To break out of a while loop, call break;
- To stop execution of statements and start again, call continue;
- All iterations can be written as while-loops.
void setup() {
    size(500, 500);
    smooth();
    float diameter = 500.0;
    while (diameter > 1.0) {
        ellipse(250, 250, diameter, diameter);
        diameter = diameter * 0.9;
    }
}
void draw() { }

void setup() {
    size(500, 500);
    smooth();
    float diameter = 500.0;
    while (true) {
        ellipse(250, 250, diameter, diameter);
        diameter = diameter * 0.9;
        if (diameter <= 1.0) break;
    }
}
void draw() { }

An aside ... Operators
+-, -, *, / and ...
i++; equivalent to i = i + 1;
i += 2; equivalent to i = i + 2;
i--; equivalent to i = i - 1;
i -= 3; equivalent to i = i - 3;
i *= 2; equivalent to i = i * 2;
i /= 4; equivalent to i = i / 4;
i % 3; the remainder after i is divided by 3 (modulo)

Iteration: for-loop
for (initialization; continuation_test; increment) {
statement;
// continue;
// break;
}
• A kind of iteration construct
• initialization, continuation test and increment commands are part of statement
• To break out of a while loop, call break;
• To stop execution of statements and start again, call continue;

void setup() {
    size(500, 500);
    smooth();
    float diameter = 500.0;
    while (diameter > 10) {
        ellipse(250, 250, diameter, diameter);
    }
}
void draw() { }

void setup() {
    size(500, 500);
    smooth();
    for (float diameter = 500; diameter > 1; diameter -= 10) {
        ellipse(250, 250, diameter, diameter);
    }
}
void draw() { }

for (int i = 0; i < 10; i++) {
    print(i);
    println();
}

for (int i = 0; i < 10; i++) {
    if (i % 2 == 1) continue;
    print(i);
    println();
}