

## for loops

- ▶ Processing has a control structure called the for loop that has the following form

```
for (init; condition; progress) {  
    // body of loop  
}
```

- ▶ This is equivalent to the following while loop

```
// init  
while (condition) {  
    // body of loop  
    // progress  
}
```

- ▶ The for loop provides a simple and compact loop with all three important loop components.

## for loop example

- ▶ while loop

```
int y = 20;
while (y <= 80) {
    line(0,y,width,y);
    y = y + 20;
}
```

- ▶ for loop equivalent

```
for (y = 20; int y <= 80; y = y + 20) {
    line(0,y,width,y);
}
```

## Compound assignment operators

- ▶ Processing has several operators that provide a combination of operator and assignment

operator	example	equivalent
<code>+=</code>	<code>x += 1</code>	<code>x = x + 1</code>
<code>-=</code>	<code>x -= 1</code>	<code>x = x - 1</code>
<code>*=</code>	<code>x *= 2</code>	<code>x = x * 2</code>
<code>/=</code>	<code>x /= 2</code>	<code>x = x / 2</code>
<code>%=</code>	<code>x %= 2</code>	<code>x = x % 2</code>
<code>++</code>	<code>x++</code>	<code>x = x + 1</code>
<code>--</code>	<code>x--</code>	<code>x = x - 1</code>

- ▶ Main advantages: less typing, enforces variable consistency, easier to change variable names

## for loop example using compound assignment operators

- ▶ for loop

```
for (y = 20; int y <= 80; y = y + 20) {  
    line(0,y,width,y);  
}
```

- ▶ equivalent for loop

```
for (y = 20; int y <= 80; y += 20) {  
    line(0,y,width,y);  
}
```

## Some fun programs

- ▶ draw a line when a row is even
- ▶ lines: red on multiples of 5, green on rows multiples of 7, blue on multiples of 11
- ▶ draw a grid of lines
- ▶ lines on rows that are powers of 2
- ▶ triangles along the diagonal