Lesson Plan Musings

Concepts

Naming (Variables)

```
name = make.componant(port number)
```

Componants:

- Button
- Motors
- Drivetrain

Actions (Functions)

```
componant.action()
componant.action(values)
make.action(values)
```

Actions:

- `button.pressed()`
- `motor.spin(power, seconds)` & `motor.stop()`
- `drivetrain.drive(power, seconds)`, `drivetrain.turn(power, seconds)`, `drivetrain.curve(left power, right power, seconds)`, `drivetrain.stop()`
- `make.wait(seconds)` & `make.wait_until(button.pressed)`
- `led_on(led_port)`, `led_off(led_port)`, & `led_blink(led_port, seconds)`

Code Order

Reads top to bottom, doesn't go to the next line until the previous one finishes.

Program Format

```
import make

name = make.componant(port)
name = make.componant(port)
name = make.componant(port)

name.action(values)
name.action(values)
make.action(values)
name.action()
name.action(values)
```

Example Lesson Code

General Program

```python
# All programs must start with:
import make

# Then, you name all of your componants:
stopbutton = make.button(1)
grabbyarm = make.smallmotor(6)
leftmotor = make.largemotor(7)
rightmotor = make.largemotor(8)
myrobot = make.drivetrain(leftmotor, rightmotor)

# Now you do all your actions
grabbyarm.spin(power=70)
make.wait_until(stopbutton.pressed)
grabbyarm.stop()

# Turn on some LEDs
make.led_on(led_port=1)
make.led_on(led_port=2)
make.led_on(led_port=3)

myrobot.drive(power=100, seconds=2)
myrobot.turn(power=-40, seconds=0.5)
myrobot.curve(left_power=90, right_power=60, seconds=5)

# Turn off the LEDs
make.led_off(led_port=1)
make.led_off(led_port=2)
make.led_off(led_port=3)

# Blink an LED quickly, another for 2 seconds, and then another quickly
make.led_blink(led_port=11)
make.led_blink(led_port=12, seconds=2)
make.led_blink(led_port=13)

make.wait(seconds=2)
myrobot.turn(power=10)
make.wait_until(stopbutton.pressed)

# Remember, code reads top to bottom! Each line won't run until the one above
# it finishes, and once the program reaches the end of the code everything
# stops
```

Common Loop Mistake

```python
import make

stopbutton = make.button(2)
motor = make.smallmotor(3)

while not stopbutton.pressed():
    motor.spin(50, 2)
    make.wait(2)
```