Week 1 - Friday

COMP 1800

Last time

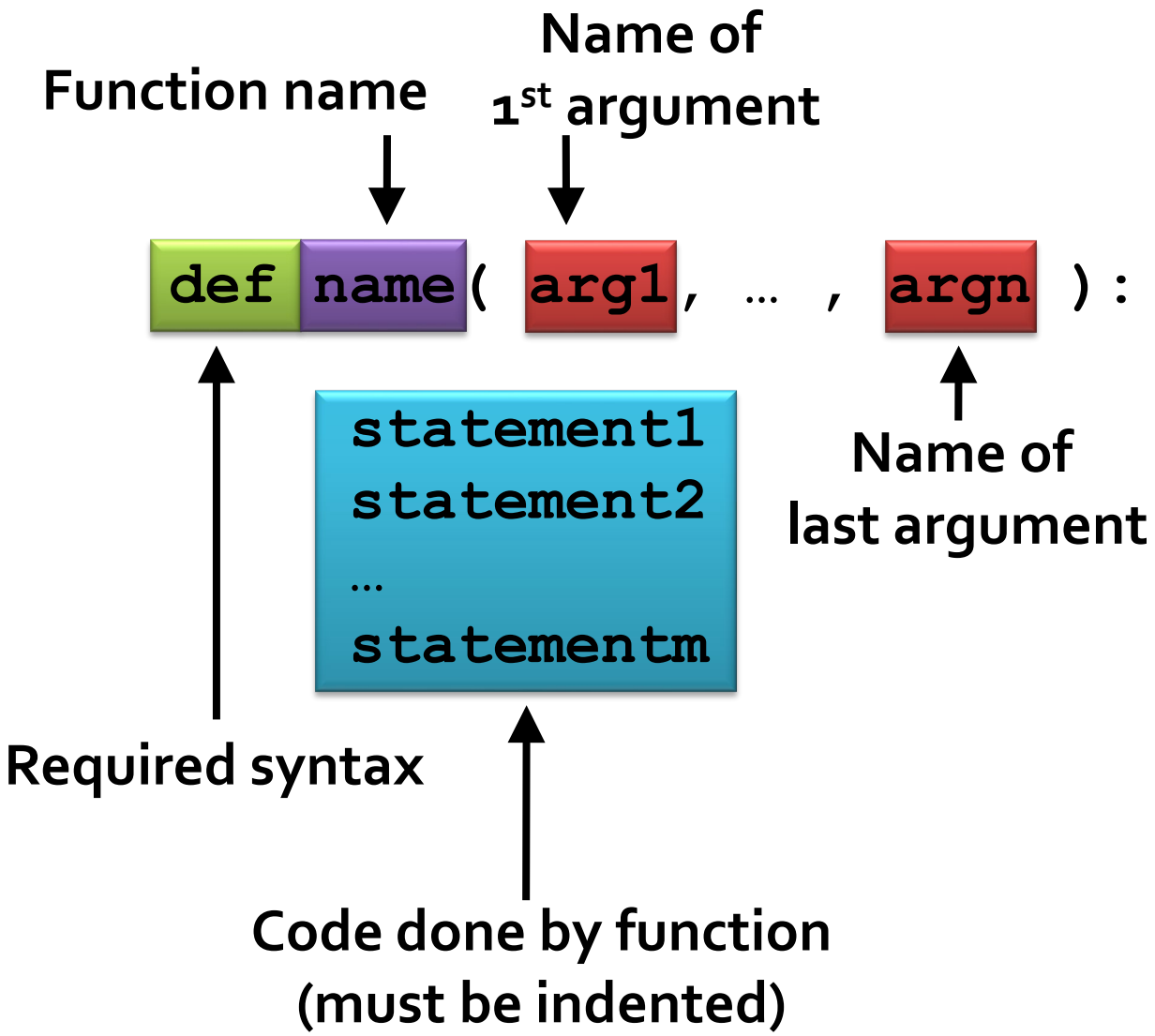
- What did we talk about last time?
- Python basics
  - Integers and floating-point numbers
  - Variables
  - Comments
  - Print statements
- Turtle

Questions?

Functions

- A really powerful tool in most programming languages is the ability to package up some code into a chunk that you can use over and over
- This idea has different names in different languages:
  - Function
  - Method
  - Subroutine
  - Procedure
- A key feature of functions is that they can take zero or more arguments that allow you to tell the function to do different things

Defining a function



Function to draw a square

- The following function takes a turtle and a side length and uses the turtle to draw a square whose sides have that length

```
def drawSquare(anyTurtle, side):
  anyTurtle.forward(side)
  anyTurtle.right(90)
  anyTurtle.forward(side)
  anyTurtle.right(90)
  anyTurtle.forward(side)
  anyTurtle.right(90)
  anyTurtle.forward(side)
  anyTurtle.right(90)
```

Calling a function

- Defining a function is just half the story
- We have to **call** the function (meaning, tell it to run) in order to use it
- In this case, the function expects a turtle and a number
- We can call it like this:

```
yertle = turtle.Turtle()
drawSquare(yertle, 50)
```

- Note that the function has to be defined *before* the code that calls it

Why are functions so great?

- We usually write functions because we want to do something many times
- For example, I could draw lots of squares with different sizes using the same function

```
yertle = turtle.Turtle()
drawSquare(yertle, 50)
drawSquare(yertle, 100)
drawSquare(yertle, 150)
```

- Functions are also good because they make code more readable, if you give them meaningful names

Turtle methods

- The book has a much longer list, but here are a few useful turtle methods

Method	Parameter(s)	Description
`forward`	Distance	Move forward
`backward`	Distance	Move backward
`left`	Angle	Turn counter-clockwise
`right`	Angle	Turn clockwise
`up`	None	Pick up the turtle's tail (to stop drawing)
`down`	None	Put down the turtle's tail (to draw again)
`heading`	None	Return the angle the turtle is pointing
`position`	None	Return the position of the turtle
`goto`	x, y	Move the turtle to (x, y)

(Better) function to draw a square

- Can we write a function that also draws a square but:
  - Lets us specify the center point
  - Also specifies the side length
- Here's the header of the function, and we'll fill in the rest:

```
def betterSquare(anyTurtle, x, y, side):
```

for loops

- Often, we want to repeat something
- The easiest way to do that in Python is with a **for** loop:

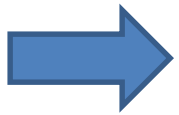
```
for i in range(n):
    statement1
    statement2
    statement3
    …
```

- All the statements in the **for** loop are repeated **n** times

Using for loops

- We'll use for loops for lots of things, but for now, we can make the original **`drawSquare`** function shorter:

```
def drawSquare(anyTurtle, side):
    anyTurtle.forward(side)
    anyTurtle.right(90)
    anyTurtle.forward(side)
    anyTurtle.right(90)
    anyTurtle.forward(side)
    anyTurtle.right(90)
    anyTurtle.forward(side)
    anyTurtle.right(90)
```



```
def drawSquare(anyTurtle, side):
    for i in range(4):
        anyTurtle.forward(side)
        anyTurtle.right(90)
```

Practice

- Our **`betterSquare`** function is a little more complicated than **`drawSquare`**, but it's still possible to make it shorter using a for loop
- Try it out on your own time!

Assignment 1

Work Day for Assignment 1

Upcoming

Next time…

- **`math`** module
- Accumulator pattern
- Computing pi

Reminders

- Read Chapter 2 of the textbook