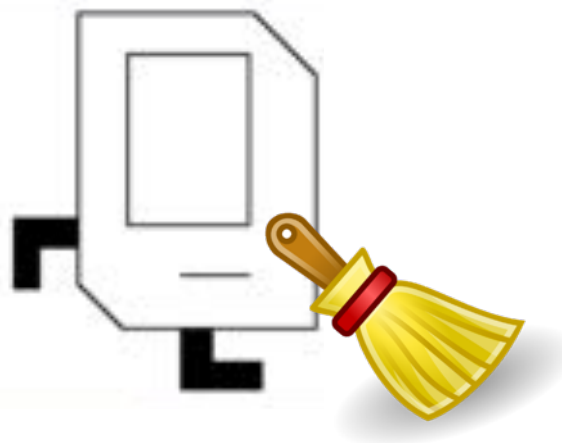




# Functions and Parameters

Chris Piech and Mehran Sahami  
CS106A, Stanford University

# Housekeeping



- Reminder: Diagnostic is next Wednesday
  - Takes place during class time
  - Like an exam
    - Really meant for you to gauge your understanding
  - Covers material through Friday
  - Email Juliette if you have a time conflict or are outside the Americas time zones



# Learning Goals

1. Get more practice with function parameters
2. Understand information flow in a program
3. Learn about Python's doctest feature



# Recall, Our Friend the Function

```
def main():    function "call"  
    avg = average(5.0, 10.2)  
    print(avg)
```

function "definition"

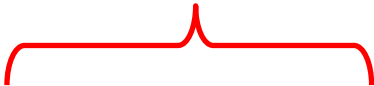
```
def average(a, b):  
    sum = a + b  
    return sum / 2
```



# Recall, Our Friend the Function

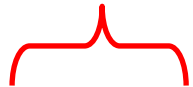
```
def main():  
    avg = average(5.0, 10.2)  
    print(avg)
```

arguments



```
def average(a, b):  
    sum = a + b  
    return sum / 2
```

parameters



# Parameters



Parameters let  
you provide a  
function with  
some information  
when you are  
calling it.



# A Full Program

```
# Constant - visible to all functions  
MAX_NUM = 4
```

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

# A Full Program

```
# Constant - visible to all functions  
MAX_NUM = 4
```

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```



Understand the mechanism

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i

0

```
def main():  
    for i in range(MAX NUM):  
        print(i, factorial(i))
```

i

0

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 0

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i

0

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n

0

result

i

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n

0

result

1

i



```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n

0

result

1

i

1

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n

0

result

1

i

1

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n

0

result

1

i

1

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

1

i

0

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

1

i

0

0 1

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 1

0 1

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i      1

0    1

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 1

0 1



```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n  result  i

0 1

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n  result  i

0 1

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n 1      result 1      i 1

0   1

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n 1      result 1      i 1

0   1

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n 1      result 1      i 1

0   1

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
  
    return result
```

n 1      result 1      i 2

0   1

```
def factorial(n):  
    result = 1  
    for i in range(1, n+1):  
        result *= i  
  
    return result
```

n 1      result 1      i 2

0   1

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

1

i

1

0 1



```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

1

i

1

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i    2

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 2

```
0 1  
1 1
```

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i

2

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i

2

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

2

i

2

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

2

i

2

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 3

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |



```
def main():  
    for i in range(MAX NUM):  
        print(i, factorial(i))
```

i 3

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 3

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 3

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

6

i

3

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

6

i

3

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |
| 3 | 6 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 4

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |
| 3 | 6 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 4

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |
| 3 | 6 |

```
def main():  
    for i in range(MAX_NUM):  
        print(i, factorial(i))
```

i 4

Done!

|   |   |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |
| 3 | 6 |



# Parameters



Every time a function is called, new memory is created for that call.

Parameter values are passed in.

All *local* variables start fresh (no old values)



An interlude:  
doctest

# Doctest

```
def factorial(n):  
    """  
    This function returns the factorial of n  
    Input: n (number to compute the factorial of)  
    Returns: value of n factorial  
    Doctests:  
    >>> factorial(3)  
    6  
    >>> factorial(1)  
    1  
    >>> factorial(0)  
    1  
    """  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
    return result
```

# Doctest

```
def factorial(n):  
    """  
    This function returns the factorial of n  
    Input: n (number to compute the factorial of)  
    Returns: value of n factorial  
    Doctests:  
    >>> factorial(3)  
    6  
    >>> factorial(1)  
    1  
    >>> factorial(0)  
    1  
    """  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
    return result
```

Say this was in file "fact.py"

To run doctests (on PC):

```
> py -m doctest fact.py -v
```

# Testing: *Why* doesn't your program work?



Let's try it!!

# Bad Times With functions

# NOTE: This program is buggy!!

```
def add_five(x):  
    x += 5
```

```
def main():  
    x = 3  
    add_five(x)  
    print("x = " + str(x))
```

Let's "trace"  
this program



# Bad Times With functions

# NOTE: This program is buggy!!

```
def add_five(x):  
    x += 5
```

```
def main():  
    x = 3  
    add_five(x)  
    print("x = " + str(x))
```





# Good Times With functions

# NOTE: This program is **feeling just fine...**

```
def add_five(x):  
    x += 5  
    return x
```

```
def main():  
    x = 3  
    x = add_five(x)  
    print("x = " + str(x))
```



# Good Times With functions

# NOTE: This program is feeling just fine...

```
def add_five(x):  
    x += 5  
    return x
```

```
def main():  
    x = 3  
    x = add_five(x)  
    print("x = " + str(x))
```



When we want to “reassign” x inside a helper function, employ the `x = change(x)` pattern!



# Parameter passing mechanism



When a parameter is passed during a function call, a **new variable** is created for the lifetime of the function call.

That new variable may or may not have the **same name** as the value that was passed in!



# No inherent connection between these two

# NOTE: This program is buggy!!

```
def add_five(x):  
    x += 5
```

These are two  
separate variables.  
They are not linked!

```
def main():  
    x = 3  
    add_five(x)  
    print("x = " + str(x))
```

Only relationship:  
value of main's x is  
used when creating  
add\_five's x





Later on in class... we will see cases where changes to variables in helper functions seem to persist! It will be great. We will let you know when we get there and exactly when that happens!



Careful!

# No Functions in Functions

```
def main():  
    print("hello world")  
    def say_goodbye():  
        print("goodbye!")
```



Technically legal, but often a sign at the start that you are confusing function *definition* and function *call*



# No functions in functions

```
def main():  
    print("hello world")  
    say_goodbye()
```

```
def say_goodbye():  
    print("goodbye!")
```





# Learning Goals

1. Get more practice with function parameters
2. Understand information flow in a program
3. Learn about Python's doctest feature





Enter year for calendar: 2021

Month #1

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  |     |     |     |     |     |     |

Month #2

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     | 1   | 2   | 3   | 4   | 5   | 6   |
| 7   | 8   | 9   | 10  | 11  | 12  | 13  |
| 14  | 15  | 16  | 17  | 18  | 19  | 20  |
| 21  | 22  | 23  | 24  | 25  | 26  | 27  |
| 28  | 29  |     |     |     |     |     |

Month #3

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  | 31  |     |     |

Month #4

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |

Month #5

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 8   | 9   | 10  | 11  | 12  | 13  | 14  |
| 15  | 16  | 17  | 18  | 19  | 20  | 21  |
| 22  | 23  | 24  | 25  | 26  | 27  | 28  |
| 29  | 30  | 31  |     |     |     |     |

Month #6

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     | 1   | 2   | 3   | 4   |
| 5   | 6   | 7   | 8   | 9   | 10  | 11  |
| 12  | 13  | 14  | 15  | 16  | 17  | 18  |
| 19  | 20  | 21  | 22  | 23  | 24  | 25  |
| 26  | 27  | 28  | 29  | 30  |     |     |

Month #7

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  |     |     |     |     |     |     |

Month #8

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     | 1   | 2   | 3   | 4   | 5   | 6   |
| 7   | 8   | 9   | 10  | 11  | 12  | 13  |
| 14  | 15  | 16  | 17  | 18  | 19  | 20  |
| 21  | 22  | 23  | 24  | 25  | 26  | 27  |
| 28  | 29  | 30  | 31  |     |     |     |

Month #9

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     | 1   | 2   | 3   |
| 4   | 5   | 6   | 7   | 8   | 9   | 10  |
| 11  | 12  | 13  | 14  | 15  | 16  | 17  |
| 18  | 19  | 20  | 21  | 22  | 23  | 24  |
| 25  | 26  | 27  | 28  | 29  | 30  |     |

Month #10

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Month #11

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  |     |     |     |

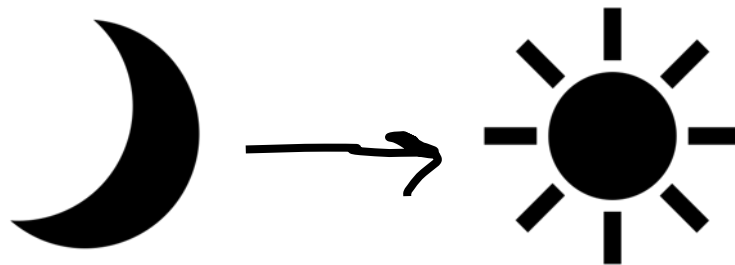
Month #12

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     | 1   | 2   | 3   |
| 4   | 5   | 6   | 7   | 8   | 9   | 10  |
| 11  | 12  | 13  | 14  | 15  | 16  | 17  |
| 18  | 19  | 20  | 21  | 22  | 23  | 24  |
| 25  | 26  | 27  | 28  | 29  | 30  | 31  |

# Brief History



Hmm, whoever is in  
charge of this lunar  
calendar has been  
slacking!



# Brief History



Ok so we can have 12 months each with exactly 30 days, and then every year ends with a 5 day festival



# Brief History



Lol no. How about  
instead each month  
has a totally  
different number of  
days?

Any my month  
(July) must have 31.  
Enjoy this random  
decision for the  
next 2,000 years!



Enter year for calendar: 2021

Month #1

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  |     |     |     |     |     |     |

Month #2

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     | 1   | 2   | 3   | 4   | 5   | 6   |
| 7   | 8   | 9   | 10  | 11  | 12  | 13  |
| 14  | 15  | 16  | 17  | 18  | 19  | 20  |
| 21  | 22  | 23  | 24  | 25  | 26  | 27  |
| 28  | 29  |     |     |     |     |     |

Month #3

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  | 31  |     |     |

Month #4

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |

Month #5

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 8   | 9   | 10  | 11  | 12  | 13  | 14  |
| 15  | 16  | 17  | 18  | 19  | 20  | 21  |
| 22  | 23  | 24  | 25  | 26  | 27  | 28  |
| 29  | 30  | 31  |     |     |     |     |

Month #6

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     | 1   | 2   | 3   | 4   |
| 5   | 6   | 7   | 8   | 9   | 10  | 11  |
| 12  | 13  | 14  | 15  | 16  | 17  | 18  |
| 19  | 20  | 21  | 22  | 23  | 24  | 25  |
| 26  | 27  | 28  | 29  | 30  |     |     |

Month #7

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  |     |     |     |     |     |     |

Month #8

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     | 1   | 2   | 3   | 4   | 5   | 6   |
| 7   | 8   | 9   | 10  | 11  | 12  | 13  |
| 14  | 15  | 16  | 17  | 18  | 19  | 20  |
| 21  | 22  | 23  | 24  | 25  | 26  | 27  |
| 28  | 29  | 30  | 31  |     |     |     |

Month #9

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     | 1   | 2   | 3   |
| 4   | 5   | 6   | 7   | 8   | 9   | 10  |
| 11  | 12  | 13  | 14  | 15  | 16  | 17  |
| 18  | 19  | 20  | 21  | 22  | 23  | 24  |
| 25  | 26  | 27  | 28  | 29  | 30  |     |

Month #10

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Month #11

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  |     |     |     |

Month #12

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     | 1   | 2   | 3   |
| 4   | 5   | 6   | 7   | 8   | 9   | 10  |
| 11  | 12  | 13  | 14  | 15  | 16  | 17  |
| 18  | 19  | 20  | 21  | 22  | 23  | 24  |
| 25  | 26  | 27  | 28  | 29  | 30  | 31  |

# Final Product

Enter year for calendar: 2021

Month #1

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  |     |     |     |     |     |     |

Month #2

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     | 1   | 2   | 3   | 4   | 5   | 6   |
| 7   | 8   | 9   | 10  | 11  | 12  | 13  |
| 14  | 15  | 16  | 17  | 18  | 19  | 20  |
| 21  | 22  | 23  | 24  | 25  | 26  | 27  |
| 28  | 29  |     |     |     |     |     |

Month #3

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  | 31  |     |     |

Month #4

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |

Month #5

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 8   | 9   | 10  | 11  | 12  | 13  | 14  |
| 15  | 16  | 17  | 18  | 19  | 20  | 21  |
| 22  | 23  | 24  | 25  | 26  | 27  | 28  |
| 29  | 30  | 31  |     |     |     |     |

Month #6

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     | 1   | 2   | 3   | 4   |
| 5   | 6   | 7   | 8   | 9   | 10  | 11  |
| 12  | 13  | 14  | 15  | 16  | 17  | 18  |
| 19  | 20  | 21  | 22  | 23  | 24  | 25  |
| 26  | 27  | 28  | 29  | 30  |     |     |

Month #7

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  |     |     |     |     |     |     |

Month #8

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     | 1   | 2   | 3   | 4   | 5   | 6   |
| 7   | 8   | 9   | 10  | 11  | 12  | 13  |
| 14  | 15  | 16  | 17  | 18  | 19  | 20  |
| 21  | 22  | 23  | 24  | 25  | 26  | 27  |

Month #9

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     | 1   | 2   | 3   |
| 4   | 5   | 6   | 7   | 8   | 9   | 10  |
| 11  | 12  | 13  | 14  | 15  | 16  | 17  |
| 18  | 19  | 20  | 21  | 22  | 23  | 24  |
| 25  | 26  | 27  | 28  | 29  | 30  |     |

Month #10

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     | 1   |
| 2   | 3   | 4   | 5   | 6   | 7   | 8   |
| 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  |
| 23  | 24  | 25  | 26  | 27  | 28  | 29  |
| 30  | 31  |     |     |     |     |     |

Month #11

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     | 1   | 2   | 3   | 4   | 5   |
| 6   | 7   | 8   | 9   | 10  | 11  | 12  |
| 13  | 14  | 15  | 16  | 17  | 18  | 19  |
| 20  | 21  | 22  | 23  | 24  | 25  | 26  |
| 27  | 28  | 29  | 30  |     |     |     |

Month #12

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     | 1   | 2   | 3   |
| 4   | 5   | 6   | 7   | 8   | 9   | 10  |
| 11  | 12  | 13  | 14  | 15  | 16  | 17  |
| 18  | 19  | 20  | 21  | 22  | 23  | 24  |



# Milestone #2

Enter year for calendar: 2021

Month #1

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  |     |     |     |     |     |     |



# Milestone #1

Enter year for calendar: 2021

Month #1

Sun Mon Tue Wed Thu Fri Sat

1  
↔

There are exactly 4  
characters of space  
for each day



# Helper Functions!

```
def first_day_of_year(year):  
    """  
    Returns the first day of the week for a given year, where  
    (0 = Sunday, 1 = Monday, ..., 6 = Saturday)  
    The formula in this function comes from http://mathforum.org/  
    >>> first_day_of_year(2020)  
    3  
    """  
    ... Some code you don't have to write ...
```



# Helper Functions!

```
def print_month_header(month):  
    """  
    Prints header for a given month in the calendar  
    """  
    print("Month #" + str(month))  
    print("Sun Mon Tue Wed Thu Fri Sat")
```



# Helper Functions!

```
def format_number(num):  
    """  
    Formats a one or two digit integer to fit in four spaces.  
    Returns a string of the formatted number string.  
    >>> format_number(12)  
    ' 12 '  
    >>> format_number(5)  
    ' 5  '  
    """  
    result = " " + str(num) + " "  
    if num < 10:  
        result = result + " "  
    return result
```



The Whole Burrito:  
calendar.py