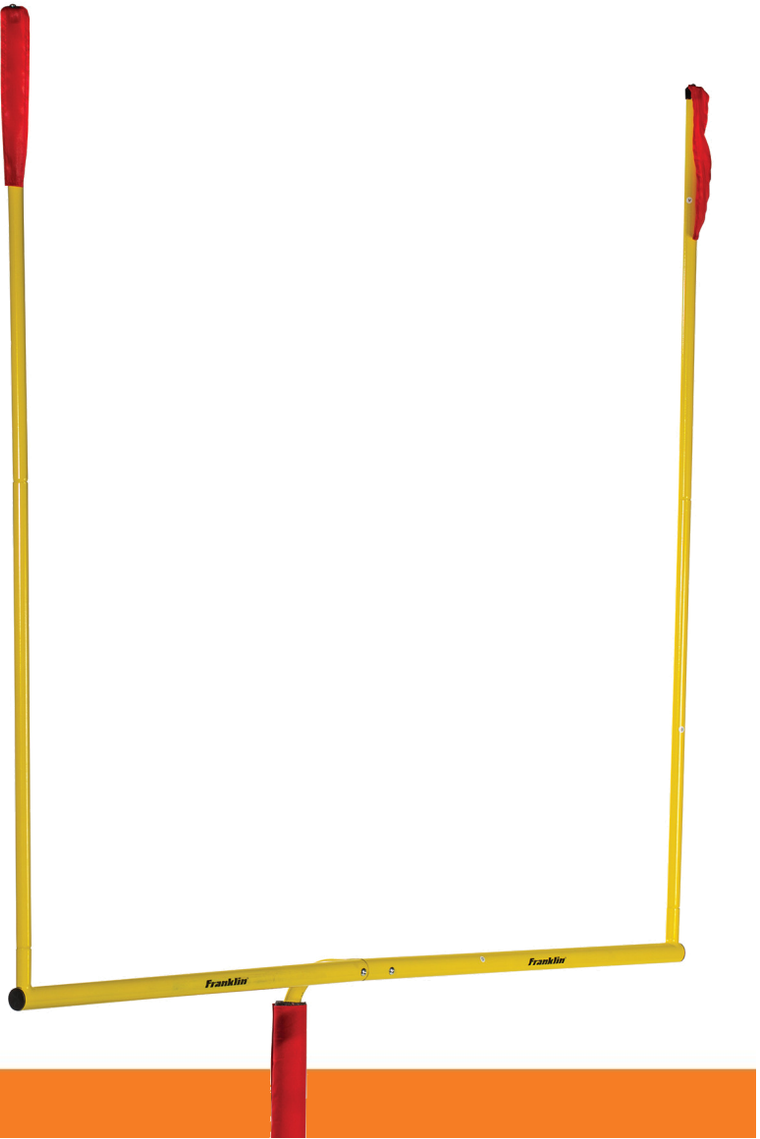
# Lists

Develop Intelligence

# Goals

1. Explain how to add an item to a list
2. Describe how to slice a list
3. Define 'list comprehension'



# Roadmap

1. Overview
2. Slices
3. Comprehension

Develop
Intelligence

# Overview

Develop Intelligence

# List



- Delimited with square brackets *[]*
- Mutable, resizable
- Holds anything

```
 # Homogenous
words = ['here','are','words']
word_count = len(words)
print(f"There's {word_count} words")

 # Heterogeneous
things = ['chicken', 3.1415, False, ['Yes', 'No']]
```

# Indexing

- Access elements with brackets *[]*
- Zero based
- Negative indexing: relative to the end

```
xs = ['it','is','like','a','weasel']
first_thing = xs[0]
second_thing = xs[1]
last_thing = xs[-1]
```

# Adding Elements

- Add single element with `append`
- Add multiple elements with `extend`
    - Or the operator +=

```
1 nums = []
2 nums.append('three')
3 nums += ['four','five']
4 nums.extend(['six','seven'])
```

# Removing Elements

- Remove by *index* with `pop`
- Remove by *value* with `remove`

```
1 bag = [1,2,3,'Chicken','Lizard']
2 bird = bag.pop(3)
3 bag.remove('Lizard')
4
5 print(bag)
```

# Membership with in

```
names = ['bloggs', 'smith', 'patel']

if 'bloggs' in names:
  print('Joe is there.')

if 'kim' not in names:
  print('Adding patel'):
  names.append('patel')
```

# Sorting

- For in-place sorting: `sort` method

```
l = [2, 1, 4, 2]
l.sort()
print(l)
```

- For a new, sorted list: `sorted`

```
l1 = [2, 1, 4, 2]
l2 = sorted(l1)
print(f"Here's l1: {l1}")
print(f"Here's l2: {l2}")
```

Develop
Intelligence

# Slices

# Overview

- Compact notation to get a subset of a sequence
- Super ***Pythonic***
- Works like the `range` function

```
xs = list(range(10))

# [start (inclusive) : end (exclusive) : jump]
first_five = xs[0:5]
last_five=xs[5:]
evens = xs[::2]
odds = xs[1::2]
```

# Recipe: Car and Cdr

- Break up a list into the head and everything else

```
xs = list(range(10))
head, tail = xs[0], xs[1:]

print(f'Head: {head}')
print(f'Tail: {tail}')
```

```
batting_order = ['Joe', 'Jane', 'Samantha', 'Maria', 'Sivi']

at_bat, waiting = batting_order[0], batting_order[1:]

print(f'{at_bat} is at bat.')
print(f'{waiting} are all in line.')
```

# Recipe: Shallow Copy


- With no numbers, a slice just makes a copy of the original

```
1 xs = list(range(10))
2 ys = xs[:]
3
4 print(f'Equals? {xs==ys}')
5 print(f'Same? {xs is ys}')
```

Develop
Intelligence

# Comprehension

Develop Intelligence

# Overview

- A ***comprehension*** is syntax for creating a new container based on an existing one
- Allows for:
  - Transformation / mapping
  - Filtration
- Similar syntax for lists, tuples, dictionaries, and sequences

# Example #1

## Old and Busted

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []

for x in fruits:
  if "a" in x:
    newlist.append(x)
```

## New Hotness

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = [x for x in fruits if "a" in x]
```

# Example #2

Develop Intelligence

## Old and Busted

```
1 ints = range(10)
2 evens = []
3 for i in ints:
4   if i % 2 == 0:
5     evens.append(i)
6
7 squared_evens = []
8 for i in evens:
9   squared_evens += [i*i]
```

## New Hotness

```
1 squared_evens = [i * i for i in range(10) if i % 2 == 0]
```

# Example #3

Develop Intelligence

## Old and Busted

```
1 word_groups = [['big','red','fast'],['run','swim','kick'], ['and','but'
2 all_words = []
3 for group in word_groups:
4   for word in group:
5     all_words.append(word)
```

## New Hotness

```
1 word_groups = [['big','red','fast'],['run','swim','kick'], ['and','but'
2 all_words = [word for group in word_groups for word in group]
```

# Lab: List Golf

# Game plan

- In VSCode, open the directory $/python_foundations/labs/
- Read instructions: src/G.O.ListGolf/README.md
- Build the applications starting with 'startingpoint'

Develop
Intelligence

# Review

1. Explain how to add an item to a list
2. Describe how to slice a list
3. Define 'list comprehension'