# CS 110
# String Indexing and Loop Tables

Adriana Picoral

# Other Announcements

- Exam 1
  - Individual Exam 1, September 21
  - Group Exam 1, September 23
    - If more than 5 mins late to group, take on your own or with others who are late
  - Review Session: Tues, Sept 20, 5-7pm (MCPRK 105)
  - Study Guide

# Review: string indexes

- Each character in a string is located at a particular **index**
- The index is zero-based

```
name = 'jeremiah'
```

```
len(name)?
```

| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| character | j | e | r | e | m | i | a | h |

# What will this print? What does it do?

```
digits = input('Type some digits: ') # '2511'
count = 0
i = 0
while i < len(digits):
    value = int(digits[i])
    count += value
    i += 1
print('count:', count)
```

# Password Validation

- Write some code that takes a string password as input, and determines if it is a “valid” password or not
- A valid password is one that:
  - Has at least one upper-case letter ( use **`isupper()`** )
  - Is at least 8 characters long
  - Has at least one of these characters:

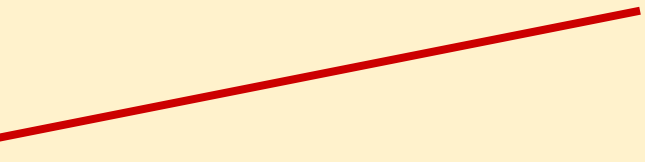
    **`! ? ;`**
- Print “valid” if valid and “not valid” if not

```
password = input('Enter a password:\n')

has_upper = False
has_special = False
i = 0
while i < len(password):
    if password[i].isupper():
        has_upper = True
    if password[i] == '!' or password[i] == '?' or password[i] == ';':
        has_special = True
    # LOCATION
    i += 1

if has_upper and has_special:
    print("Valid Password")
else:
    print("Invalid password.")
```

Activity

# Loop Table for LOCATION

```
one = 'the lost world'
two = 'the last stride'
i = min(len(one), len(two)) - 1
count = 0
while i >= 0:
    if one[i] == two[i]:
        count += 1
    i -= 1
    # LOCATION
print('tally:', count)
```



| i | count |
| --- | --- |
| | |
| | |
| | |
| | |
| | |
| | |

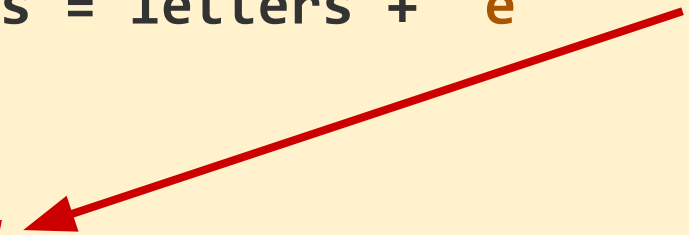
# Loop Table for LOCATION

```
one = 'the lost world'
two = 'the last stride'
i = min(len(one), len(two)) - 1
count = 0
while i >= 0:
    if one[i] == two[i]:
        count += 1
    i -= 1
    # LOCATION
print('tally:', count)
```



| i | count |
|---|---|
| 12 | 1 |
| 11 | 1 |
| 10 | 2 |
| 9 | 2 |
| 8 | 2 |
| ... | ... |

```
letters = ''
a = 0
b = 8
c = 'r'
while a < b:
    if a > 2:
        letters = letters + c
    else:
        letters = letters + 'e'
    a += 1
    b -= 1
    # LOCATION
```



Write down the value of variables `a`, `b`, and `letters` when the body of the loop ends each iteration

Do so using a loop table

```
letters = ''
a = 0
b = 8
c = 'r'
while a < b:
    if a > 2:
        letters = letters + c
    else:
        letters = letters + 'e'
    a += 1
    b -= 1
    # LOCATION
```

| a | b | letters |
|---|---|---|
| 1 | 7 | 'e' |
| 2 | 6 | 'ee' |
| 3 | 5 | 'eee' |
| 4 | 4 | 'eeer' |

**Activity**

```
i = 0
other = 100
while i < 3:
    if other > i:
        print('other > i')
        other += 4
    j = 0
    while j < 2:
        # LOCATION
        print(i, j, other)
        j += 1
    i += 1
```

Write down the value of variables `i`, `j` and `other` when the body of the loop ends each iteration

Do so using a loop table

```
i = 0
other = 100
while i < 3:
    if other > i:
        print('other > i')
        other += 4
    j = 0
    while j < 2:
        # LOCATION
        print(i, j, other)
        j += 1
    i += 1
```

| i | j | other |
|---|---|---|
| 0 | 0 | 104 |
| 0 | 1 | 104 |
| 1 | 0 | 108 |
| 1 | 1 | 108 |
| 2 | 0 | 112 |
| 2 | 1 | 112 |