Python for Analytics: Week 2

Data Type Conversion and Control Flow

JFK Example
The `range` Function and Methods of Strings

Example

Write a program to print the initials of a name given by the user. For example, if the given name is "John Fitzgerald Kennedy", the printed out message is "JFK".

```
In [17]: name = input("Key in a name: ")

         initials = name[0]                        # Get the first name initial
         while True:
             index = name.find(' ')                # Position of the first space
             if index == -1:
                 break                              # Break if no space found
             initials = initials + name[index+1]   # Add the letter after space
             name = name[index+1:]                  # Get a subset of name

         print(initials)

         Key in a name: John Fitzgerald Kennedy
         JFK
```

Methods of Strings

name ➡️

J	o	h	n		F	i	t	z	g	e	r	l	d		K	e	n	n	e	d	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

initials ➡️

J
0

In [17]:
```python
name = input("Key in a name: ")

initials = name[0]                          # Get the first name initial
while True:
    index = name.find(' ')                  # Position of the first space
    if index == -1:
        break                               # Break if no space found
    initials = initials + name[index+1]     # Add the letter after space
    name = name[index+1:]                   # Get a subset of name

print(initials)
```
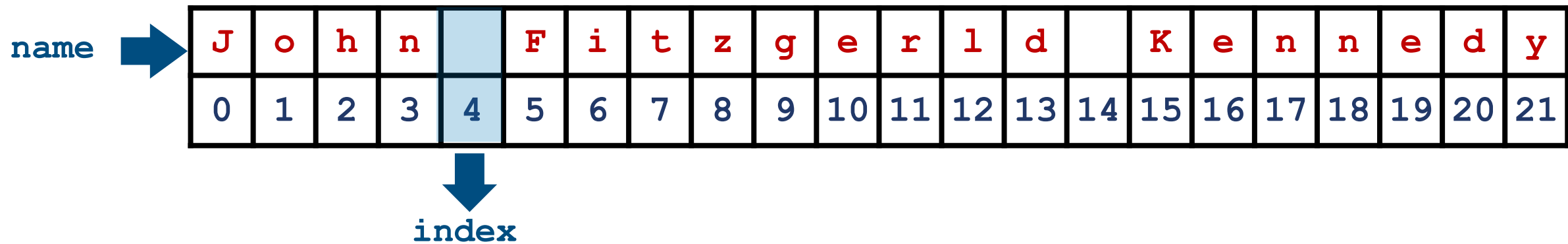
Key in a name: John Fitzgerald Kennedy
JFK

Methods of Strings

name →

J	o	h	n		F	i	t	z	g	e	r	l	d		K	e	n	n	e	d	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

↓

index

initials →

J
0

```
In [17]: name = input("Key in a name: ")

         initials = name[0]                          # Get the first name initial
         while True:
             index = name.find(' ')                  # Position of the first space
             if index == -1:
                 break                                # Break if no space found
             initials = initials + name[index+1]      # Add the letter after space
             name = name[index+1:]                    # Get a subset of name

         print(initials)

Key in a name: John Fitzgerald Kennedy
JFK
```

Methods of Strings

J	o	h	n		F	i	t	z	g	e	r	l	d		K	e	n	n	e	d	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

name

name[index+1]

initials

J	F
0	1

```
In [17]:   name = input("Key in a name: ")

           initials = name[0]                    # Get the first name initial
           while True:
               index = name.find(' ')            # Position of the first space
               if index == -1:
                   break                          # Break if no space found
               initials = initials + name[index+1]   # Add the letter after space
               name = name[index+1:]             # Get a subset of name

           print(initials)
```
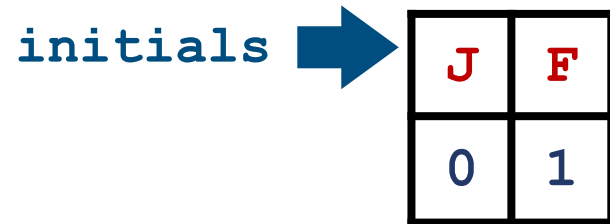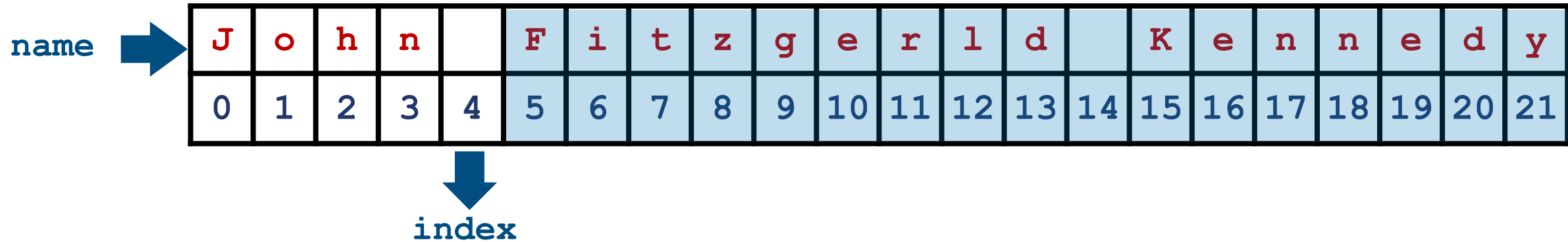
Key in a name: John Fitzgerald Kennedy
JFK

Methods of Strings

name →

J	o	h	n		F	i	t	z	g	e	r	l	d		K	e	n	n	e	d	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

↓

index

initials →

J	F
0	1

```
In [17]:    name = input("Key in a name: ")

            initials = name[0]                        # Get the first name initial
            while True:
                index = name.find(' ')                # Position of the first space
                if index == -1:
                    break                             # Break if no space found
                initials = initials + name[index+1]   # Add the letter after space
                name = name[index+1:]                 # Get a subset of name

            print(initials)
```

Key in a name: John Fitzgerald Kennedy
JFK

Methods of Strings

name ➡️

F	i	t	z	g	e	r	l	d		K	e	n	n	e	d	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

initials ➡️

J	F
0	1

In [17]:
```python
name = input("Key in a name: ")

initials = name[0]                          # Get the first name initial
while True:
    index = name.find(' ')                  # Position of the first space
    if index == -1:
        break                               # Break if no space found
    initials = initials + name[index+1]     # Add the letter after space
    name = name[index+1:]                   # Get a subset of name

print(initials)
```
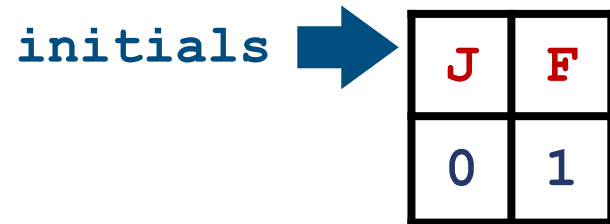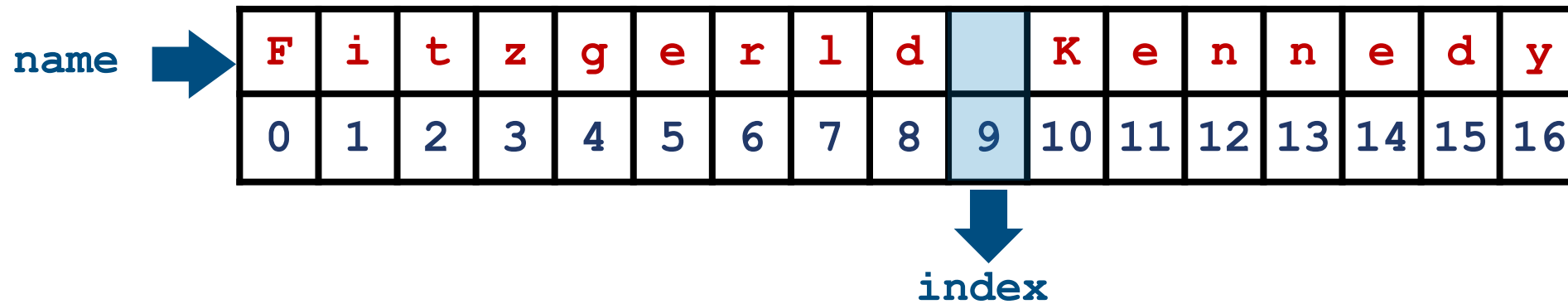Key in a name: John Fitzgerald Kennedy
JFK

Methods of Strings

name →

F	i	t	z	g	e	r	l	d		K	e	n	n	e	d	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

↓

index

initials →

J	F
0	1

```python
In [17]:  name = input("Key in a name: ")

          initials = name[0]                      # Get the first name initial
          while True:
              index = name.find(' ')              # Position of the first space
              if index == -1:
                  break                           # Break if no space found
              initials = initials + name[index+1] # Add the letter after space
              name = name[index+1:]               # Get a subset of name

          print(initials)

Key in a name: John Fitzgerald Kennedy
JFK
```

Methods of Strings

F	i	t	z	g	e	r	l	d		K	e	n	n	e	d	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

name →

`name[index+1]`

initials →

J	F	K
0	1	2

```
In [17]:  name = input("Key in a name: ")

          initials = name[0]                       # Get the first name initial
          while True:
              index = name.find(' ')               # Position of the first space
              if index == -1:
                  break                             # Break if no space found
              initials = initials + name[index+1]   # Add the letter after space
              name = name[index+1:]                 # Get a subset of name

          print(initials)

          Key in a name: John Fitzgerald Kennedy
          JFK
```

Methods of Strings

F	i	t	z	g	e	r	l	d		K	e	n	n	e	d	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

index

initials

J	F	K
0	1	2

```
In [17]:   name = input("Key in a name: ")

           initials = name[0]                      # Get the first name initial
           while True:
               index = name.find(' ')              # Position of the first space
               if index == -1:
                   break                            # Break if no space found
               initials = initials + name[index+1] # Add the letter after space
               name = name[index+1:]                # Get a subset of name

           print(initials)

           Key in a name: John Fitzgerald Kennedy
           JFK
```

Methods of Strings

name ➡

K	e	n	n	e	d	y
0	1	2	3	4	5	6

-1 ⬇

Cannot find a whitespace in **name**, so return -1

initials ➡

J	F	K
0	1	2

```
In [17]:  name = input("Key in a name: ")

          initials = name[0]                        # Get the first name initial
          while True:
              index = name.find(' ')                # Position of the first space
              if index == -1:
                  break                              # Break if no space found
              initials = initials + name[index+1]   # Add the letter after space
              name = name[index+1:]                 # Get a subset of name

          print(initials)

          Key in a name: John Fitzgerald Kennedy
          JFK
```

Methods of Strings

name ➡️

K	e	n	n	e	d	y
0	1	2	3	4	5	6

-1
⬇️

Cannot find a whitespace in name, so return -1

initials ➡️

J	F	K
0	1	2

```
In [17]:  name = input("Key in a name: ")

          initials = name[0]                      # Get the first name initial
          while True:
              index = name.find(' ')              # Position of the first space
              if index == -1:
                  break                           # Break if no space found
              initials = initials + name[index+1]  # Add the letter after space
              name = name[index+1:]               # Get a subset of name

          print(initials)

          Key in a name: John Fitzgerald Kennedy
          JFK
```

Reviewing Strings

- Strings are sequences of characters

  ‣ Length, indexing, and slicing

- Iterate characters in strings

  ‣ Direct iterations

  ‣ Iteration via `range` indexing

- String methods

  ‣ Syntax of calling methods

  ‣ Case conversion methods, `find`, and others