



Accelerating Information Technology Innovation

<http://aiti.mit.edu>

Nigeria Summer 2012
Lecture 2 – Control Structures

Agenda

- Goal
 - Break up “How are you” into “How”, “are”, “you”
- Learn about
 - If...Else
 - Loops
 - While
 - For
 - For...Else
 - Break
 - Continue
 - Nested Loops

Control Flow

Do it Once with If...Elif...Else

```
def printFirstLetter(words):  
    if words[0].isalpha():  
        print words  
    elif words[0].isspace():  
        print "space"  
    else:  
        pass
```

```
>>> printFirstLetter("Hello")
```

```
Hello
```

```
>>> printFirstLetter(" ")
```

```
space
```

```
>>> printFirstLetter("555")
```

Loop

```
def parsewords_while(words):  
    currentword = ""  
    index = 0  
    while index < len(words):  
        if words[index].isalpha():  
            currentword += words[index]  
        elif words[index].isspace():  
            print currentword  
            currentword = ""  
        else:  
            pass  
        index += 1  
    else:  
        print currentwords
```

```
>>> parsewords_while("How are y555ou")
```

```
How are
```

```
>>> parsewords_while("How are y555ou")
```

```
How are you
```

Loops: Break vs Continue

```
def parsewords_while(words):
    currentword = ""
    index = 0
    while index < len(words):
        if words[index].isalpha():
            currentword += words[index]
        elif words[index].isspace():
            print currentword
            currentword = ""
        else:
            break
        index += 1
    else:
        print currentwords
```

```
>>> parsewords_while("How are y555ou")
How are
```

```
def parsewords_while(words):
    currentword = ""
    index = 0
    while index < len(words):
        if words[index].isalpha():
            currentword += words[index]
        elif words[index].isspace():
            print currentword
            currentword = ""
        else:
            continue
        index += 1
    else:
        print currentwords
```

```
>>> parsewords_while("How are y555ou")
How are you
```

Loops

While...Else

```
def parsewords_while(words):
    currentword = ""
    index = 0
    while index < len(words):
        if words[index].isalpha():
            currentword += words[index]
        elif words[index].isspace():
            print currentword
            currentword = ""
            index += 1
    else:
        print currentword

>>> parsewords_while("How are y555ou")
How are you
```

For...Else

```
def parsewords_for(words):
    currentword = ""
    for index in range(len(words)):
        if words[index].isalpha():
            currentword += words[index]
        elif words[index].isspace():
            print currentword,
            currentword = ""
        else:
            print currentword

>>> parsewords_for("How are y555ou")
How are you
>>> range(5)
[0, 1, 2, 3, 4]
>>> range(2, 6)
[2, 3, 4, 5]
>>> range(-10, 5, 2)
[-10, -8, -6, -4, -2, 0, 2, 4]
```

Loops

For...Else (Alternate)

```
def parsewords_for(words):
    currentword= ""

    for char in words:
        if char.isalpha():
            currentword += char
        elif char.isspace():
            print currentword,
            currentword = ""

    else:
        print currentword

>>> parsewords_for("How are y555ou")
How are you
```

For...Else

```
def parsewords_for(words):
    currentword = ""

    for index in range(len(words)):
        if words[index].isalpha():
            currentword += words[index]
        elif words[index].isspace():
            print currentword,
            currentword = ""

    else:
        print currentword

>>> parsewords_for("How are y555ou")
How are you
```

Nested Loops

```
def parsewords_for(words):  
    currentword = ""  
    for char in words:  
        if char.isalpha():  
            currentword += char  
        elif char.isspace():  
            for letter in currentword:  
                print letter,  
            currentword = ""  
        else:  
            print currentword  
  
>>> parsewords_for("How are y555ou")  
H ow a re you
```

Why we need control structures

- Decide what to do next
- Do certain actions for certain events
- Repeat a series of actions
- Break a series of actions