



Selection The Preside of Computing Using Python: 3<sup>rd</sup> Editors, 3<sup>a</sup> Princh & Enbody, Copyright & 2017 Presson Education, Inc.



































Besides selecting which statements to execute, a fundamental need in a program is repetition.
 epeat a set of statements under some conditions.
 with both selection and repetition, we have the two most necessary programming statements.



while Loop
 condition test at top (pretest)

 test the boolean before running
 test the boolean before each iteration of the loop

 while boolean expression:

 suite





Code Listing 2.8



















"The Practice of Computing Using Python", 3rd Edit Punch & Enbody, Copyright © 2017 Pearson Edut



top\_num\_str = input("What is the upper number for the range:")

# sum the divisors of number # classify the number based on its divisor sum

top\_num = int(top\_num\_str)

while number <= top\_num:

number += 1

number=2

37

39



Code Listing 2.13

Examine a range of

numbers

40

39

"The Practice of Computing Using Python", 3rd E Punch & Enbody, Copyright © 2017 Pearson Ed





















Chained Comparisons
• in Python, chained comparisons work just
like you would expect in a mathematical
expression
- given myInt has the value 5
0 <= myInt <= 5 → True
0 < myInt <= 5 < 1 → False</pre>







Truth Tables						
	р	q	not p	p and q	p or q	
-	True	True				
	True	False				
	False	True				
	False	False				
and the second			т R	he Practice of Computin Inch & Enbody, Copyrigh	g Using Python", 3 <sup>ut</sup> Edition", <b>55</b> t © 2017 Pearson Education, Inc.	
55						

**Truth Tables** not p | p and q | p or q р q True True False True False False True False True False False True 56 "The Practice of Computing Using Python", Punch & Enbody, Copyright © 2017 Pearso

56

Truth Tables						
	р	q	not p	p and q	p or q	
	True	True		True		
	True	False		False		
	False	True		False		
	False	False		False		
and the second			т Рі	he Practice of Computing Inch & Enbody, Copyright	Using Python", 3 <sup>rd</sup> Edition", <b>57</b> © 2017 Pearson Education, Inc.	

57

Truth Tables						
	р	q	not p	p and q	p or q	
	True	True	False	True	True	
	True	False	False	False	True	
	False	True	True	False	True	
	False	False	True	False	False	
and the second se				"The Practice of Con Punch & Enbody, Co	nputing Using Python", 3 <sup>uii</sup> Edition", <b>59</b> pyright © 2017 Pearson Education, Inc.	

Truth Tables						
_F	C	q	not p	p and q	p or q	
-	True	True			True	
-	True	False			True	
F	alse	True			True	
F	alse	False			False	
1	The Practice of Computing Using Python", 3 <sup>rd</sup> Edition", 58 Punch & Enbody, Copyright © 2017 Pearson Education, Inc.					



## Precedence and Associativity

 relational operators have precedence and associativity just like numerical operators

	Operator	Description	
	0	Parenthesis (grouping)	
	**	Exponentiation	
	+x, -x	Positive, Negative	
	*,/,%	Multiplication, Division, Remainder	
	+,-	Addition, Subtraction	
	<, <=, >, >=,!=, ==	Comparisons	
	not x	Boolean NOT	
	and	Boolean AND	
	or	Boolean OR	
TABLE 2.2	Precedence of Relation	nal and Arithmetic Operators: High	est to Lowest
a dillo			
		"The Practice of Computing Using Py Punch & Enbody, Copyright © 2017 F	thon", 3 <sup>rd</sup> Edition", 61 Pearson Education, Inc.

61







<form><form><form>

More on Assignments































Developing a while Loop working with the *loop control variable* • initialize the variable, typically outside of the loop and before the loop begins • the condition statement of the while loop involves a boolean using the variable

• modify the value of the control variable during the course of the loop

"The Practice of Computing Using Python", 3rd Edition", 83 Punch & Enbody, Copyright © 2017 Pearson Education, Inc.





































Algorithm while the number does not equal one • if the number is odd, multiply by 3 and add 1 • if the number is even, divide by 2 • use the new number and reapply the algorithm

102





2 number\_str = input("Enter a positive integer:") 3 number = int(number\_str) + count = 0 % print("Starting with number:",number)
7 print("Sequence is: ", end=' ') > while number > 1: # stop when the sequence reaches 1 if number%2: # number is odd
 number = number\*3 + 1 else: # number is even number = number/2 print (number, ", ", end=' ') # add number to sequence # add to the count count +=1 19 else: print() # blank line for nicer output
print("Sequence is ",count," numbers long") 105 "The Practice of Computing Using Pyti Punch & Enbody. Copyright © 2017 Pe



