Week 3: Exceptions

Common Ecross	Le sor - Same -
	with how you typed something
- Name Error - Variable example	print (" Hello, World)
	zno closing " results in syntax error
	instead of float, int instant of Pluat etc.
Fo [®]	ivense Co he report
How To Write Excep	
	phons:
Try: # try to run code be	neath
→ # Code	
Except: Value Error: # 2	Except == any error Value Error specifies
> # Run this code if error or	r specified what error it will run code for
Else:	error
# if it doesn't seleve on e	error, run this code optional break after
V	
of the below	
Coac Below	
cleanes bouls to les	are loop + return to return value
C130 030 01and 10 100	
D -1) 1 bodle	exception without doing caything with the
Jass - Used to manche	Deception without wenty Edyling was 1
AT SURE CONTROL OF THE PARTY OF	is more it ie. don't print anything or close
program etc.	
	Exectly
	Except Nalve Error:

Pass

Recap zimilal aniproduction
-Point
Syntax Error - You typed something in wrons. (shower) respected -
Example: print ("Hellot, Morla) sond restolant al
le No closing quotes result in syntax error
and the second s
Writins exceptions:
try:
X = int(input("What's x?"))
except Value Franci
except Value Error: Print ("Thats not an integer")
elected to the sear the med colors to be brained many land
···· Print (f" x is {x}")
Value Error - input is a string or float or integer at a strine that
it wasn't supposed to be. my training
example: Entering a string or a float in the above code block
and from roading many share of Normans Scope.
Pass - A way to handle an exception silently. Without printing on
anything or closing Program etc.
-> except Value Error:
···· Pass
The state of the s
Debussing - Method used for you to determine why certain code isnt working as interded and
code iont working as invention
ie. using print to have the computer display what its doing

	Debuggins Methods
	-U-Tat
	- Debogger (Vs code)
	4 Includes heark points
	The state of the s
1	Libraries - Functions pre-made given to you by python. North
	a defalt finction such as print but more so a cuer.
1	Mude a ready by someone else lon use a policy my one men
1	to import it at the top of your code . 3372 301 AV
	ie: module (reserving no ton stad) trint
	Line I impart random It random being the name of a liorally
	Line 2: (" {x } 2: x " 3) + 1179
D	Line 3
D	
D	Instead of importing an entire module you can imports only
	section of it by using import from at the top of your come
	a Section of it by using import from at the top of your cone
	a Section of it by using import from at the top of your cone
000	ie: Line 1 i from random in port choice # Narrows scope
0000	ie: Line 1 : from random in port choice # Narrows scope Line 2 :
0000	ie: Line 1 i from random import choice # Narrows scope Line 2 i
0000	Line 3 i Line 3
0000	a Section of it by Using import from at the top of your code ie: Line 1 i from random import choice # Narrows scope Line 2 i Line 3 i * Useful Random Notes
0000	a Section of it by Using import from at the top of your code ie: Line 1 i from random import choice # Narrows scope Line 2 i * Useful Random Notes = appearance returns input: to alist
0000	a Section of it by Using import from at the top of your code ie: Line 1 i from random import choice # Narrows scope Line 2 i * Useful Random Notes - append returns input to alist ie:
0000	a Section of it by Using import from at the top of your condice: Line 1 i from random import choice # Narrows scope Line 2 i Line 3 i * Useful Random Notes - append returns input to alist ie
	a Section of it by Using import from at the top of your code ie: Line 1 i from random import choice # Narrows scope Line 2 i * Useful Random Notes - append returns input to alist ie:
	a Section of it by Using import from at the top or your cone ic: Line 1 : from random import choice # Narrows scope Line 2 : Line 3 : * Useful Random Notes - append returns input to alist ie grades = [] stores to list
	a Section of it by Osins import from at the top of your cone ic: Line 1 i from random inport choice # Narrows scope Line 2 i Line 3 i * Useful Random Notes - append returns input to alist ic of the top of your cone grades = [] stores to list arade = float (input many) out and of the top of your cone arade = float (input many) out and of the top of the
	a Section of it by Using import from at the top or your cone ic: Line 1 : from random import choice # Narrows scope Line 2 : Line 3 : * Useful Random Notes - append returns input to alist ie grades = [] stores to list
	a Section of it by Osins import from at the top of your cone ic: Line 1 i from random inport choice # Narrows scope Line 2 i Line 3 i * Useful Random Notes - append returns input to alist ic of the top of your cone grades = [] stores to list arade = float (input many) out and of the top of your cone arade = float (input many) out and of the top of the