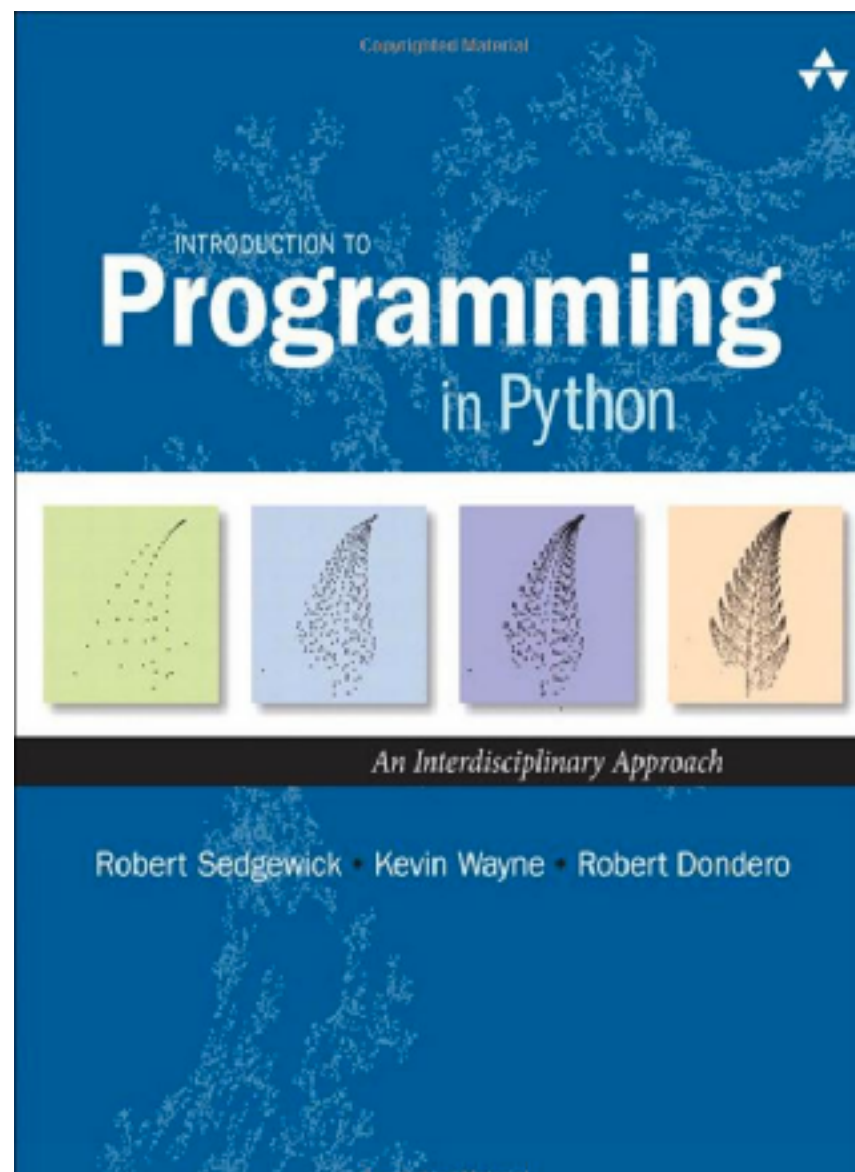


Parte I: Intro pensamiento computacional

Clase 01: Variables y tipos de datos

Diego Caro
dcaro@udd.cl



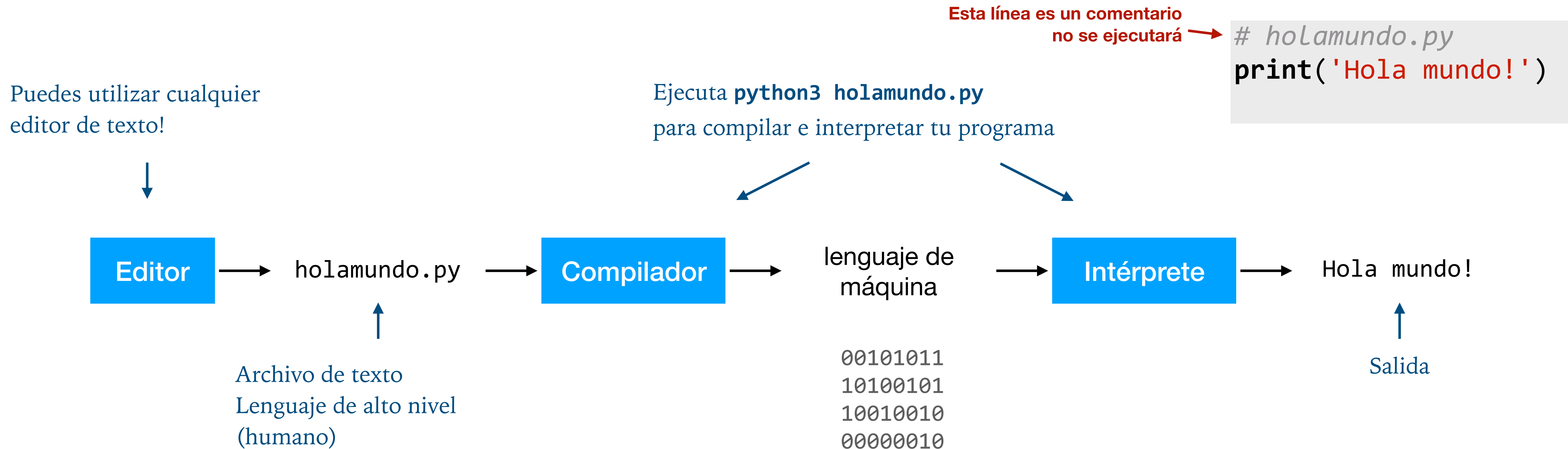
Basada en presentaciones oficiales de libro Introduction to Programming in Python (Sedgewick, Wayne, Dondero).

Disponible en <https://introcs.cs.princeton.edu/python>

Objetivos de la clase

- Reconocer el concepto de variable y tipo de datos
- Identificar tipos de datos numéricos y de texto
- Utilizar operadores de aritmética básicos
- Reconocer uso de mecanismo para entrada/salida de datos

Escribiendo un programa



```
2. bash  
$ python3 holamundo.py  
Hola mundo!  
$
```

```
Command Prompt  
C:\Users\diegomon\Documents>python holamundo.py  
Hola mundo!  
C:\Users\diegomon\Documents>
```

Memoria

- Dispositivo electrónico que permite almacenar datos durante un intervalo de tiempo. Ejemplos:
 - memoria RAM (memoria de acceso directo, toma el mismo tiempo acceder a cada lugar de la memoria)
- Los números, letras y todo lo que vemos se almacenan en memoria!
 - Ejemplo de memoria muy pequeña



1	3	4	4	6	7	0	1
---	---	---	---	---	---	---	---

Sintáxis: primer programa en Python

- Sintáxis: reglas de cómo escribir en algún lenguaje/idioma.
 - El español tiene reglas... Por ejemplo:
 - Python tiene su propia sintaxis!

La casa verde
artículo sustantivo adjetivo

Comentario → 1 *# Hacker en Halt and Catch Fire*

2 **Identificador** nombre **Operador (asignación)** = **Literal** 'Cameron Howe'

3 **Identificador** print **Argumento** (nombre)

P: ¿Qué es una **variable**?
R: Memoria que permite almacenar y consultar datos!

nombre: 'Cameron Howe'



Cameron Howe, programadora protagonista de la serie Halt and Catch Fire.

Entrada y salida de datos

Entrada de texto

```
nombre = input('Indica tu nombre: ')\nprint('Tu nombre es', nombre)
```

Entrada de números

```
nombre = input('Indica tu nombre: ')\nedad = int(input('Indica tu edad: '))\nprint('Hola', nombre, 'veo que tienes', edad, 'años')\ndecenas = edad // 10\nprint('Y al menos tienes', decenas, 'decadas')
```

- Funciones de entrada/salida:
 - input(...) recibe datos por teclado
 - print(...) imprime datos en la pantalla
- Funciones de cambios de tipo:
 - int(...) transforma texto en número
 - str(...) transforma número a texto

P: ¿Qué sucede si eliminamos la función int en la línea 2 de la entrada de números?

R: Error en línea 4, no puedes hacer una división con un texto!

Operaciones sobre tipos de datos

- Operaciones en tipos de datos string (texto):
 - Concatenar (unir): +
 - Agregar al final: +=
- Operaciones en tipos de datos numéricos:
 - Aritmética: +, -, /, *, %, //
 - Y también se puede usar +=, -=, *=, etc...

values	integers						
typical literals	1234 99 0 1000000						
operations	sign	add	subtract	multiply	floored divide	remainder	power
operators	+ -	+	-	*	//	%	**

Python's int data type

values	real numbers				
typical literals	3.14159 6.022e23 2.0 1.4142135623730951				
operations	addition	subtraction	multiplication	division	exponentiation
operators	+	-	*	/	**

Python's float data type

Manejo de texto

```
1 primero = input('Indica tu nombre: ')\n2 segundo = input('Indica tu apellido: ')\n3 nombre = primero + ' ' + segundo\n4 print('Tu nombre es', nombre)
```

Operaciones aritméticas

```
1 # En la siguiente línea importamos la función raíz cuadrada.\n2 # Revisa otras funciones matemáticas\n3 # en https://docs.python.org/3/library/math.html\n4 from math import sqrt\n5\n6 n = float(input('Ingrese un número decimal: '))\n7 print('n = ', n)\n8 print('5 por n = ', 5*n)\n9 print('n dividido dos = ', n/2)\n10 print('raíz cuadrada de n = ', sqrt(n))
```

Nota: la operación/funcionalidad se realiza dependiendo del tipo de dato asociado (ej: entero o string).

Conversión de tipos de datos

- La función input por defecto representa variables de tipo str (texto).
- Podemos convertir str en números usando:
 - `int(...)` para números enteros
 - `float(...)` para números decimales
- Y también podemos convertir números en texto, usando `str(...)`
- **Nota:** la función `type(variable)` indica el tipo de dato de una variable

```
1 amigos = input('Cuántos amigos tienes en facebook? ')
2 comun = input('Cuántos amigos tienes en común con tu mejor amiga? ')
3
4 num_amigos = int(amigos)
5 num_comun = int(comun)
6 porcentaje = num_comun/num_amigos*100
7 print('Wow, tienes', porcentaje, '% de amigos en común con tu mejor amiga!')
8
9 print('tipo variable amigos:', type(amigos))
10 print('tipo variable num_amigos:', type(num_amigos))
```

P: Qué sucede si no usamos la conversión `int(...)` ?

R:

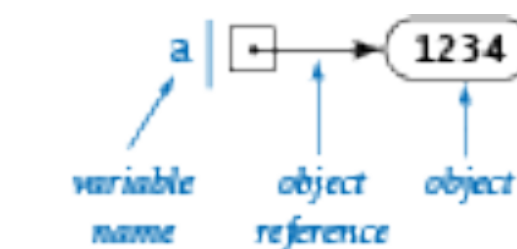
```
Cuántos amigos tienes en facebook? 12
Cuántos amigos tienes en común con tu mejor amiga? 1
Traceback (most recent call last):
  File "conversion-bad.py", line 4, in <module>
    porcentaje = num_comun/num_amigos*100
TypeError: unsupported operand type(s) for /: 'str' and 'str'
```


Un poco más de sintaxis

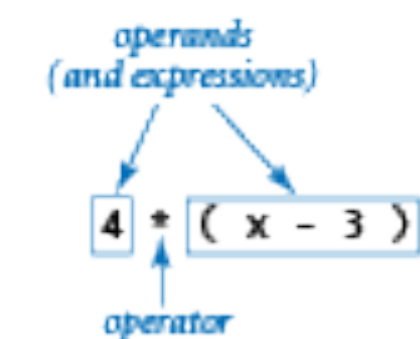
- **Literal:** código que representa un tipo de dato.
 - "hola" es un texto (tipo de dato str)
 - 3.14 es un número float
- **Identificador:** nombre que se le puede dar una variable o función.
 - nombre, print,
 - Hay algunas reglas que debemos cumplir...
- **Variables:** es una referencia a una parte de la memoria que representa algún dato. La usamos para guardar resultados parciales a medida que la computación se realiza
- **Expresiones:** combinación de variables, literales e identificadores.
- **Traza:** evolución de variables a medida que se ejecuta un programa.

type	set of values	common operators	sample literals
int	integers	+ - * // % **	99 12 2147483647
float	floating-point numbers	+ - * / **	3.14 2.5 6.022e23
bool	true-false values	and or not	True False
str	sequences of characters	+	'AB' 'Hello' '2.5'

Basic built-in data types



A variable refers to an object



Anatomy of an expression

	a	b	c
a = 1234	1234		
b = 99	1234	99	
c = a + b	1234	99	1333

Your first informal trace

Sintáxis: identificadores

- Deben comenzar con una letra o con un guión bajo
 - Pueden contener dígitos, y guión bajo
 - x, nombre_apellido, y55, _holahola
 - Pero no pueden comenzar con números o contener caracteres de operacion:
 - 12, 1x, -55, 33mineros, micro\$oft
- Tampoco pueden ser palabras reservadas:
 - Ejemplo: int, if, return, float...
- Ejemplos:
 - edad; suma_parcial;
 - nombre; temperatura;
- Deben ser informativos!
 - Siglas son confusas: LAX, CCP, myvar
 - No use nombres largos:
mi_variable_de_tipo_string
- Buenos ejemplos:
 - process_type, is_moving, current_time
- **Nota:** Recuerde que su código será leído por otra persona... Inclúido su “futuro yo”

Palabras reservadas en Python

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Actividad

- Escribe un programa que imprima:

```

**      ***      *****      **      *      **
**      ***      **      **      **      ***      **
**      ***      **      **      **      ***      **
**      ***      **      **      **      ***      **
*****      **      **      **      **      **      **
**      ***      **      **      **      **      **
**      ***      **      **      **      **      **
**      ***      **      **      ***      ***      **
**      ***      *****      *      *

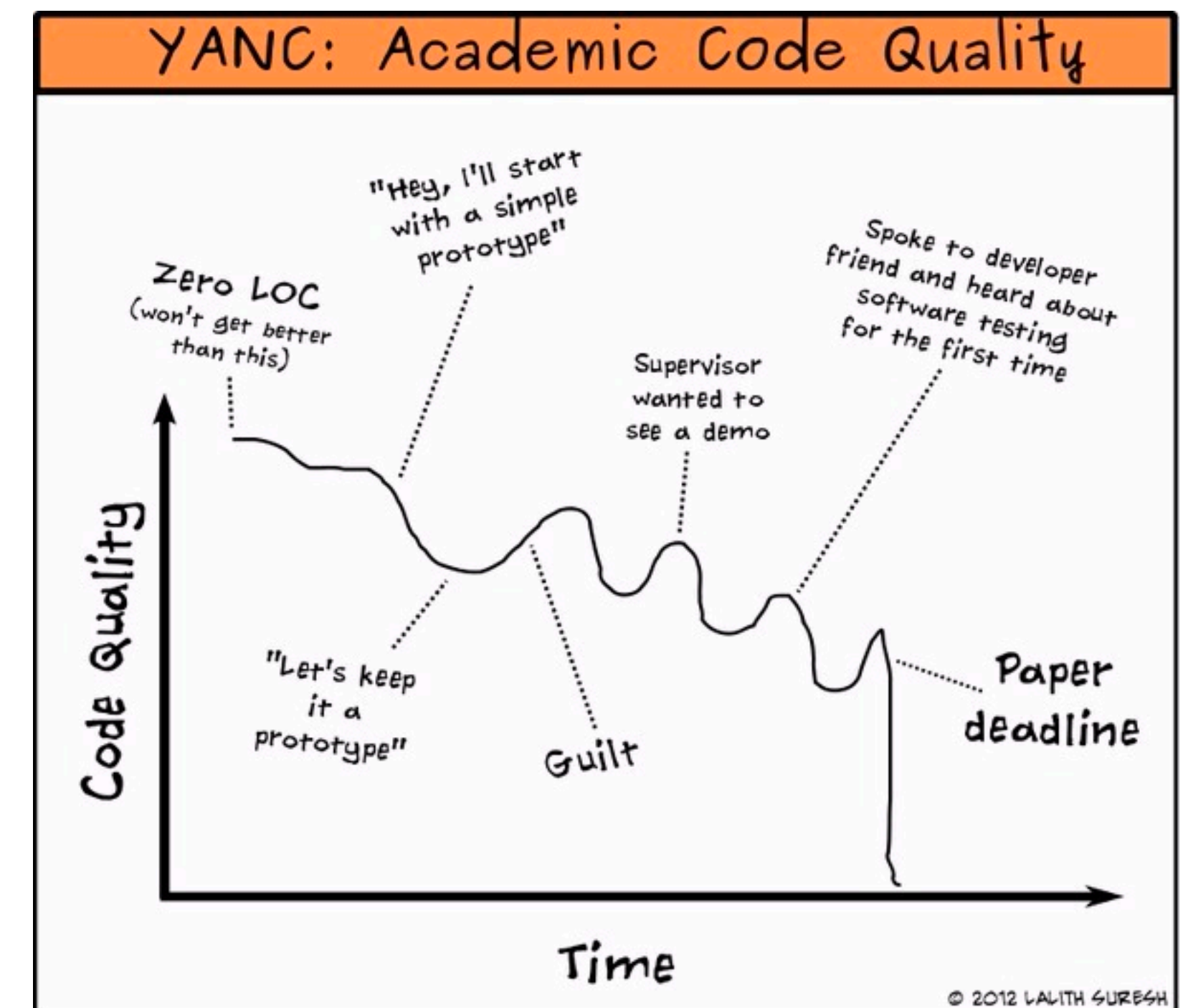
```

- Crea un programa que calcule la solución de una ecuación de segundo grado de la forma $ax^2 + bx + c = 0$
 - Asume que el usuario entrega el valor de a, b y c.
 - Recuerda que la raíz se puede calcular con $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
 - Mira el archivo `aritmetica.py`
- Modifica la segunda línea del archivo `entero.py`
 - ¿Qué sucede? ¿Qué significa el error?
- Revisa el programa `argumento.py`
 - Desafío: Ejecutalo desde la consola
 - Desafío: Modifica el programa para que imprima 3 nombres



Diana Carrier
@artemis_134

Programming is like magic. You write very specific instructions in arcane languages to invoke commands, and if you get it even a little bit wrong you risk unleashing demons and destroying everything.



Resumen

Funciones

- **input(...)**: captura de datos desde el teclado
- **print(...)**: imprimir en pantalla
- **type(...)**: verificar el tipo de dato de una variable o literal

Funciones matemáticas

```
>>> import math
>>> dir(math)
['__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

Más info en <https://docs.python.org/3/library/math.html>

Conceptos

- **concatenar**: unir dos variables de tipo str
- **entrada/salida**: ingresar y extraer datos

Sintáxis

```
var_texto = 'literal con text'
var_entero = -112
var_float = 3.1416

#concatenar
t1 = 'mi casa'
t2 = 'es bonita'
texto = t1 + ' ' + t2 # 'mi casa es bonita'

#operaciones aritméticas
suma = 1+2 # -, *, /
div = 1 // 2 # division entera
mod = 1 % 2 # resto
```