Python PCAP 31-03 Exam Practice Questions List (Difficult or Reminder)

by naufalnashif

Harga Sertifikasi PCAP 31: $295 (+- Rp. 4.697.432,50) | link

Courses:

- Cisco Python Essential 1 (Course Entry Level PCEP 30 + Discount 20%)
- Cisco Python Essential 2 (Course Associate PCAP 31 + Discount 50%)
- Udemy PCAP 31-03

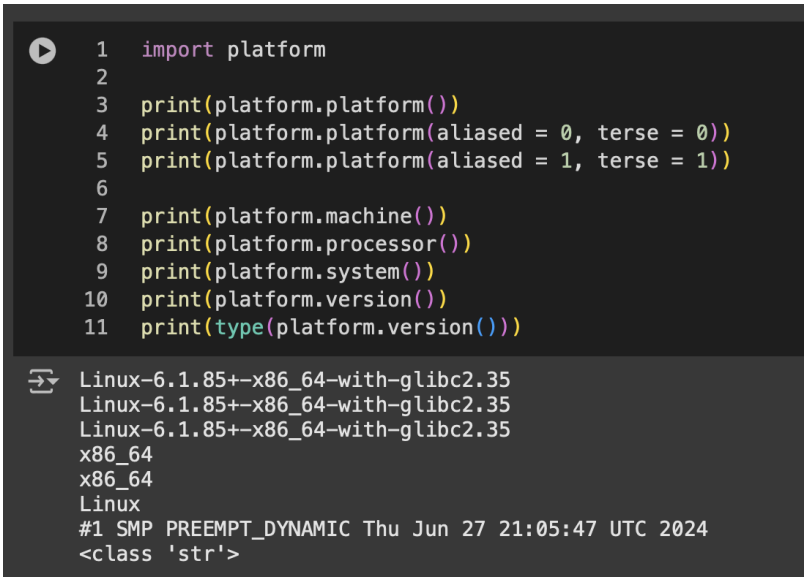
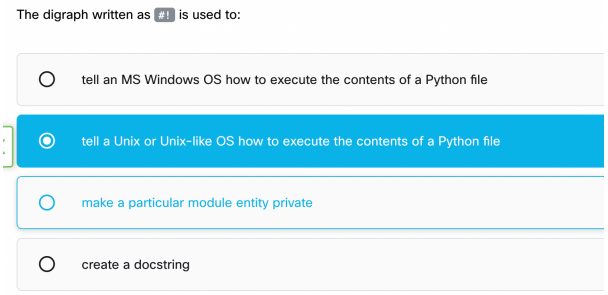
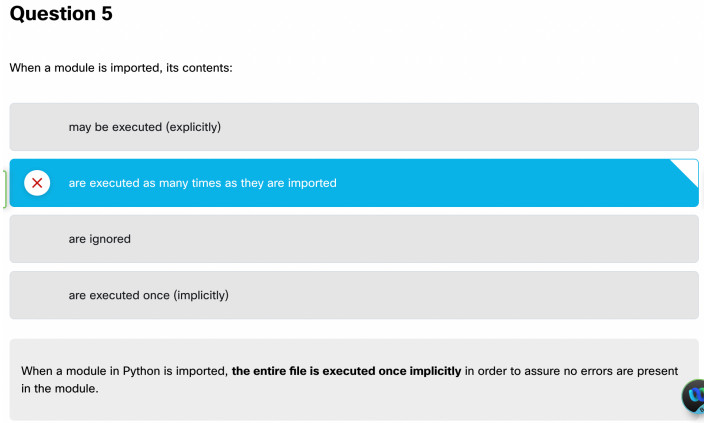
Exam Practice:

- Exam Practice Without Answer
- Exam Practice With Answer

> *Note :*
> *Passing grade 70/100 dari 40 soal, atau 28/40 soal. Berarti maksimal salah 12 soal.*

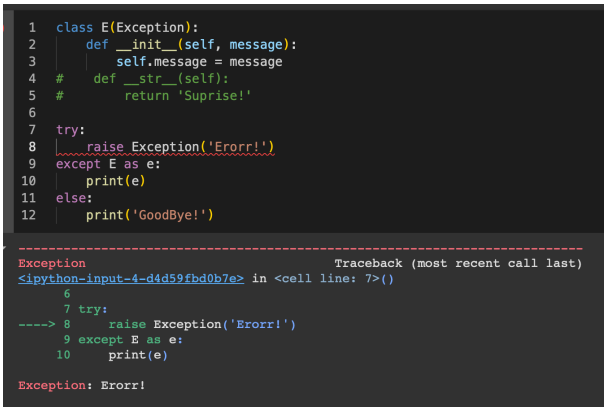
Modules & Packages

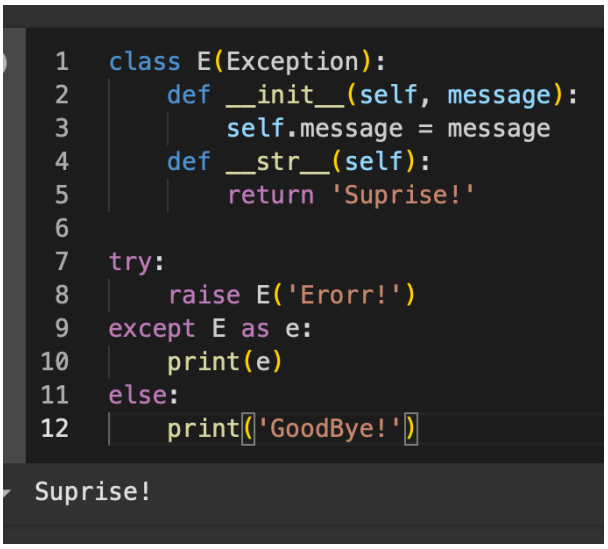
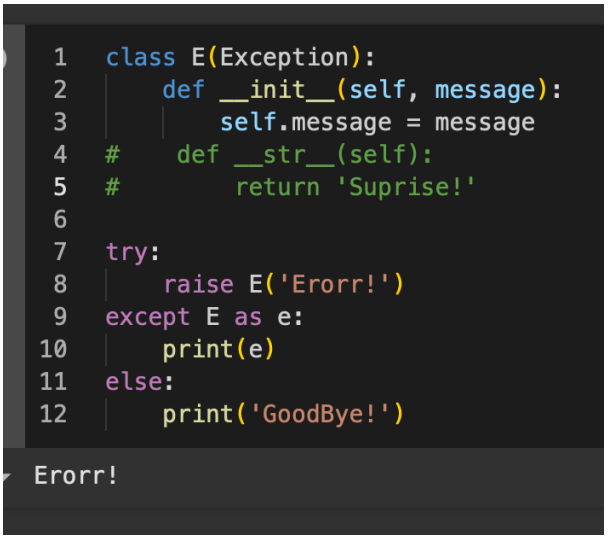
Soal	Keterangan
Select all valid option(s) about the result of dir() [ ] A list of filenames inside the directory [ ] A list of the module's attribute [ ] A list of names of class attributes [ ] A list of names of object attributes [ ] A list of names of the base class attributes Answer >>>	
`math.ceil() or math.floor()`	return `int`
`math.factorial(x)`	`x (int) >= 0` `0! = 1`
`random.choices(populasi, weights, k)`	return `list`
`random.sample(populasi, k)`	return `list`

Soal	Keterangan
	
	shebang or hashbang memberi tahu path to interpreter untuk dijalankan di Unix OS
	ketika import module, maka hanya di eksekusi sekali walaupun kita melakukan import berulang ulang
	- the console/terminal - the screen

Error Handling *(Exceptions)*

Soal	Keterangan

What is the output of the following code?
class E (Exception):
  def __init__ (self, message):
  self.message = message
  def __str__ (self):
  return "Surprise"
 	
try :
  raise Exception("Stop")
except E **as** e:
  print(e)
else :
  print("Goodbye")
( ) Unhandled Exception
( ) Surprise
( ) Stop
( ) Goodbye
	Unhandled Exception :
	ada `raise` yang tidak di tangkap
	
	Jika tidak ada error di blok `try`, baru `else` akan tereksekusi
	 

 	# `__str__` akan dipanggil ketika `print(e)` alih - alih memanggil `self.message` yang berisi pesan `Error!`
 	

Soal	Keterangan

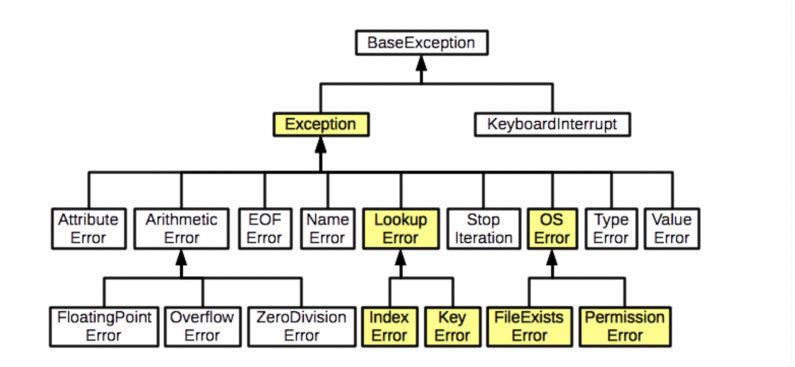
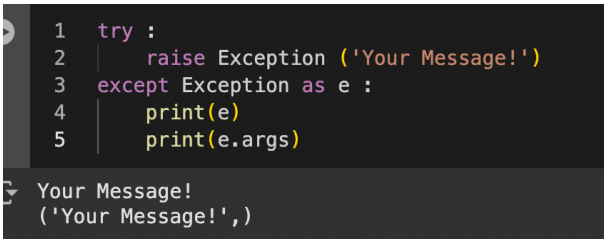
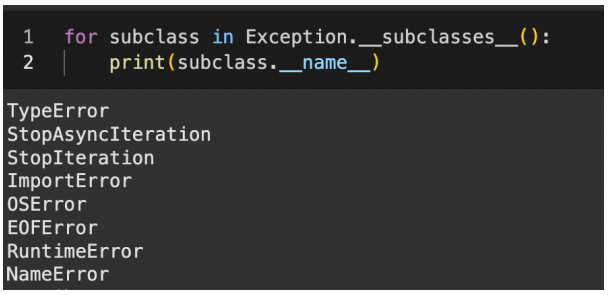
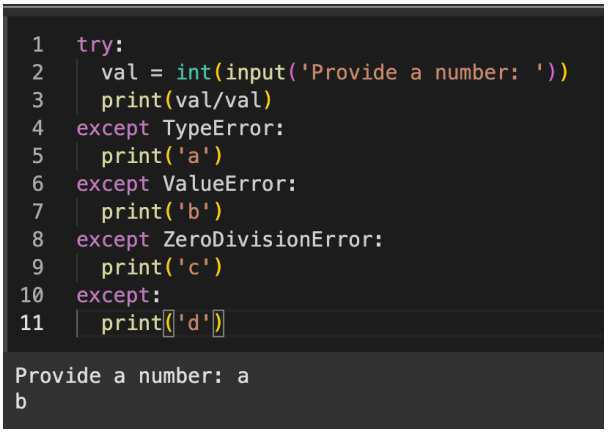
What is the output of the following code?
`try :`
`  raise Exception`
`except :` Default Except must be last
`  print( "Spam" , end= " )`
`except BaseException:`
`  print( "Ham" , end= " )`
`except Exception:`
`  print( "Eggs" )`
	Default except must be last

`class SpamException (Exception):`
`  def __init__ (self, message):`
`  <<< INSERT CODE HERE >>>`
`  self.message = message`
`raise SpamException( "Spam" )`
[X] super().__init__(message)
[ ] Exception.__init__(self, message)
[X] super(SpamException, self).__init__(message)
[ ] super.__init__(message)
	pemanggilan attr pada kelas induk
	1. `super().__init__(message)`
	2. `super(SpamException, self).__init__(message)`
	3. `ParentClass.__init__(self, message)`

What is the result of the following code? `>>> assert ( False , 'Trigger Assertion' )` [ ] No output [ ] Trigger Assertion [ ] SyntaxError: invalid syntax [ ] Assertion is always true	tuple yang memiliki isi dianggap `True` `assert False, 'Message'` akan selalu `False`
Which of the following are examples of built-in **concrete** Python exceptions? (Select **two** answers) ☐ `BaseException` ☐ `ArithmeticError` ☑ `ImportError` ☑ `IndexError`	build in konkret (error plaing dasar/spesifik)

`try:`
`  raise Exception`
`except BaseException:`
`  print("a")`
`except Exception:`
`  print("b")`
`except:`
`  print("c")`
`a`
	`BaseException` merupakan exception paling luas sehingga exception di bawahnya tidak akan ter eksekusi

`try:`
`except Exception:`
`except BaseException:`
`except:`
	sebaiknya dalam membuat urutan exception dari exception yang paling spesifik baru ke yang paling luas
	dan untuk `except` harus di akhir

Soal	Keterangan
	
	`e.args` -> tuple
	menghasilkan hierarchy error. Cara buat exception hanya melakukan inheritance dari error paling bawah
	`self.args` akan mengoverride msg di instance
	type sudah benar `str` hanya value yang salah

Strings

Soal	Keterangan

What is the output of the following code?
`print( "C:\Program Files\Microsoft\Windows NT" , end= "" )`
`print( "\")`
( ) Syntax Error
( ) C:\Program Files\Microsoft\Windows NT\
( ) Replace escaped characters with "?" e.g. C:?rogram Files?icrosoft?indows NT?
( ) Ignore escaped characters e.g. C: rogram Filesicrosoftindows NT
	Answer :
	- Syntax Error

* ASCII , UNICODE, UTF-8, codepoints, escape sequences Select all valid option(s) below about string [ ] string.ascii_letters is a concatenation of ascii_lowercase and ascii_uppercase [ ] string.ascii_letters is a concatenation of ascii_lowercase, ascii_uppercase and digits [ ] string.ascii_letters are all printable characters found in the keyboard [ ] string.ascii_lowercase contains 'abcdefghijklmnopqrstuvwxyz' [ ] string.ascii_uppercase contains 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'	Strings : upper + lower bukan digit dan any char

What is the output of the following code?
`spam = chr( 'a' )`
`ham = ord(spam)`
`print(spam, ham)`
( ) 97 a
( ) TypeError: an integer is required (got type str)
( ) TypeError: chr() takes exactly two arguments (1 given)
( ) TypeError: ord() takes exactly two arguments (1 given)
( ) Syntax Error
	Answer : b
	`chr(int)` not `chr('a')`

`spam = 'FuBar'`
`ham = spam[:]`
[ ] spam == ham
[ ] id(spam) == id(ham)
[ ] spam.startswith(ham)
[ ] spam.endswith(ham)
[ ] spam.equals(ham)
	Jika *immutable*, python ga akan membuat memori baru
	`id(spam) == id(ham)`

What is the output of the following code? `>>> None * 2` ( ) 0 ( ) None ( ) NoneNone ( ) TypeError: unsupported operand type(s) for * Answer >>>	not support NoneType
What is the output of the following code? `>>> sorted([ 5 , "1" , 100 , "34" ])` ( ) ["1", 5, "34", 100] ( ) [5, "1", "34", 100] ( ) ["1", "100", "34", "5"] ( ) [1, 5, 34, 100] ( ) TypeError: '<' not supported between instances of 'str' and 'int'	cannot sort `int` and `str`

What is the output of the following code?
spam.txt
`spam ham eggs`
spam.py
`f = open( 'spam.txt' , 'r' )`
`if 'eggs' in f:`
`  print( 'Eggs found' )`
`else :`
`  print( 'Eggs not found' )`
( ) Eggs found
( ) Eggs not found
( ) TypeError: argument type TextIOWrapper not iterable
( ) SyntaxError: invalid syntax
	open(file, 'r') return file object not file content
	
	content = f.read()

Which of the calls below are valid String function calls and will return True?
[ ] 'abc123'.isalnum()
[ ] 'abc'.isalpha()
[ ] '123abc'.isidentifier()
[ ] '123abc'.islower()
[ ] '123'.isdigit()
[ ] 'Abc'.istitle()
	`isalnum()` -> `isalpha()` + `isnumeric()`
	`isnumeric` -> termasuk `isdigit(0-9)`
	
	`isidentifier()` -> cek apakah valid sebagai variable

What is the output of the following code? `>>> sorted([ 'banana' , 'pear' , 'grapes' , 'apple' ], key= lambda x: x[:: -1 ])`	`sorted(iter, key)` key = fungsi `key = lambda x : x  -1` -> reverse setiap element

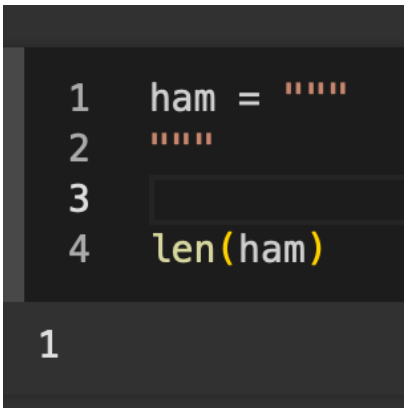
Soal	Keterangan
* .sort(), sorted(), .index() , .find(), .rfind() What is the output of the following code? `>>> "Spam Ham Eggs" .index( 'Spam' , 1 )` ( ) Spam ( ) 0 ( ) 1 ( ) ValueError: substring not found ( ) TypeError: index() takes 1 argument (2 given)	`.index(str, start, end)` 'Spam' ada di index 0 bukan 1 jadi `ValueError = not found`

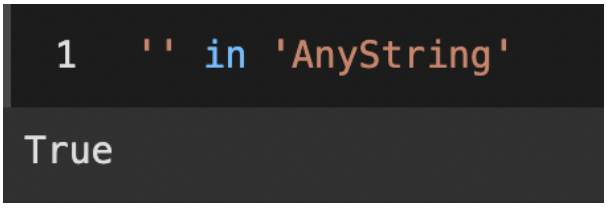
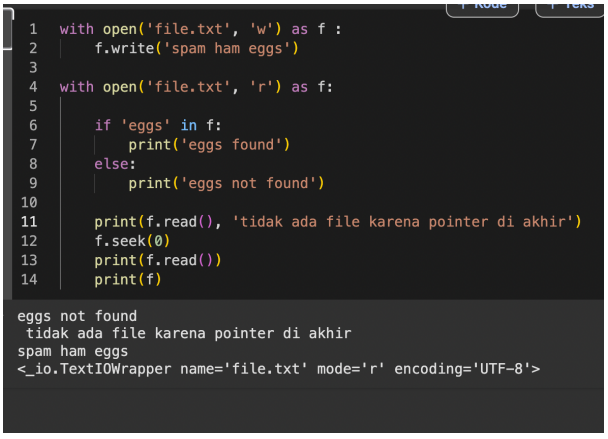
* .sort(), sorted(), .index(), .find() , .rfind()
What is the output of the following code?
`t = "Spam Ham"`
`print(t.find( "Ham" , 0 ) == t.index( "Ham" , 0 ))`
`print(t.find( "Eggs" , 0 ) == t.index( "Eggs" , 0 ))`
( ) True True
( ) True False
( ) True will be printed followed by ValueError: substring not found
( ) True will be printed followed by TypeError: find() takes 1 argument (2 given)
Answer >>>
* .sort(), sorted(), .index(), .find(), .rfind()
What is the output of the following code?
`t = "Spam Ham"`
`print(t.rfind( "am" ) == t.find( "am" ))`
`print(t.rfind( "am" , 3 ) == t.find( "am" , 3 ))`
`print(t.rfind( "am" , -3 ) == t.find( "am" , -3 ))`
( ) False False False
( ) False True True
( ) True True True
( ) True will be printed followed by TypeError: rfind takes 1 argument (2 given)
Answer >>>
	.index() -> jika tidak ditemukan maka ValueError
	.find() & .rfind() -> return -1

UTF-8 is: a synonym for *byte* a form of encoding Unicode code points the 9th version of the UTF standard a Python version name	`UTF – 8` is `a form of encoding Unicode code points`

`print('Anre' > "Anrea")`
`False`
`print('2Anre' > "1Anrea")`
`True`
`'z' < 'aaaa'`
`False`
	pembandingan string berdasarkan nilai UNICODE karakter demi karakter dari kiri, ketika menemukan perbedaan, python tidak melanjutkan pembandingan karakter berikutnya
	
	- karena `ord('2') > ord('1')` ini langsung memunculkan output `True` tanpa membandingkan karakter berikutnya
	
	- `ord('z') > ord('a')`

Soal	Keterangan

What will be printed in the following code?
`spam = """"""""`
`ham = """"`
`""""`
`print(spam, ham)`
( ) Syntax Error
( ) Two empty strings
( ) An empty string and a new line character
( ) Two new line character
Answer >>>
	

`s = 'Python'` ( ) print(s[0] + s[-1]) ( ) print(s[::5]) ( ) print(s[::-5]) ( ) print(s[::-1][::-5]) Answer >>>	step `-5` berarti membaca dari index terakhir dengan step 5 -> `nP` sedangkan step 5 -> `Pn`
	string kosong itu termasuk kedalam string apapun
	`open() as f` ini mengembalikan `object` bukan `content`, jadi tidak bisa diiterasi untuk melakukan iterasi lakukan `read` dahulu

What is the output of the following code?
`>>> "/" .join({ "Month" : "12" , "Day" : "25" , "Year" : "2021" })`
( ) 12/25/2021
( ) Month/Day/Year
( ) Month/12/Day/25/Year/2021
( ) TypeError: can only join an iterable
Answer >>>
What is the output of the following code?
`>>> "XYZ" .join( "123" )`
( ) XYZ123
( ) 123XYZ
( ) 1XYZ2XYZ3
( ) X123Y123Z
( ) TypeError: can only join an iterable
	.join pada dict hanya akan mengambil `key`

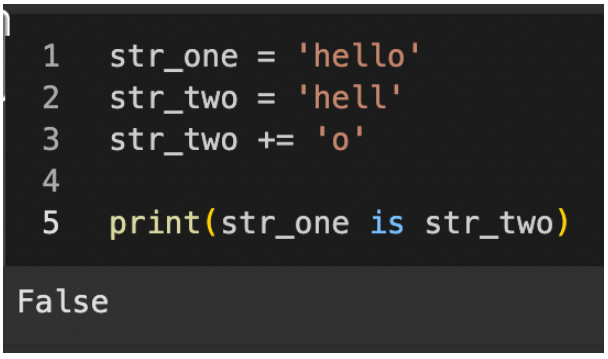
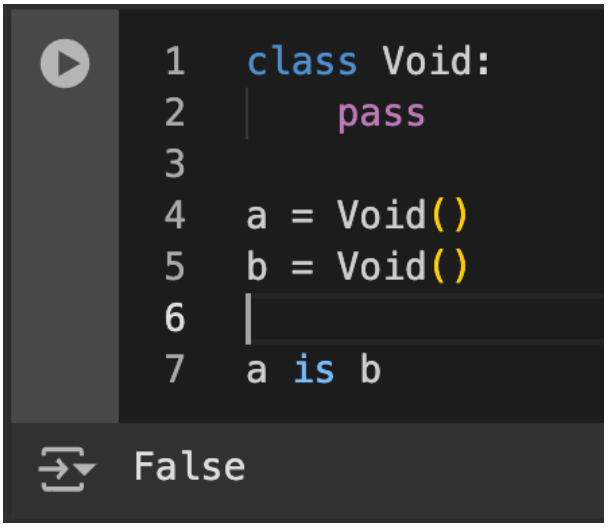
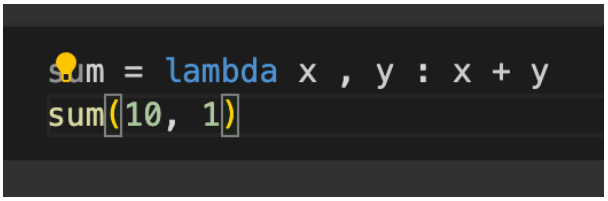
	menghasilkan string kosong dan panjang list sebanyak `/`

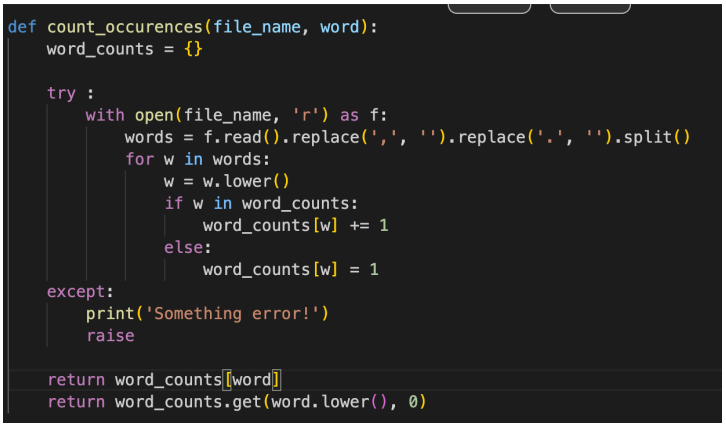
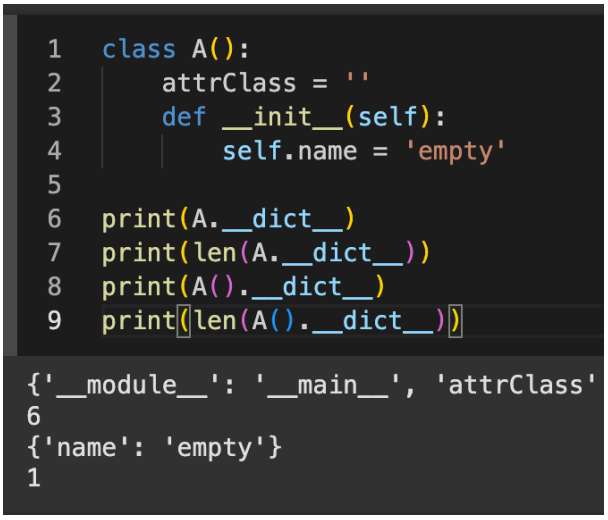
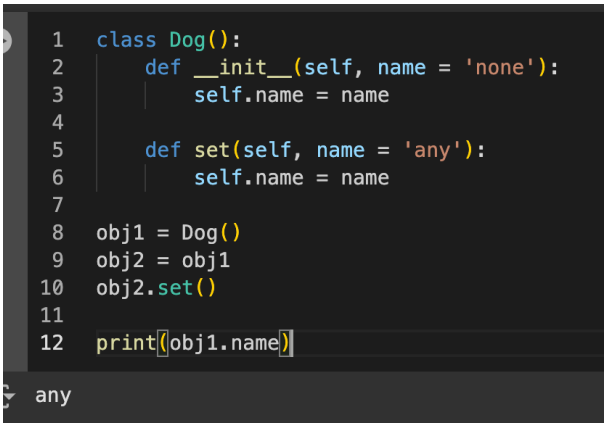
OOP

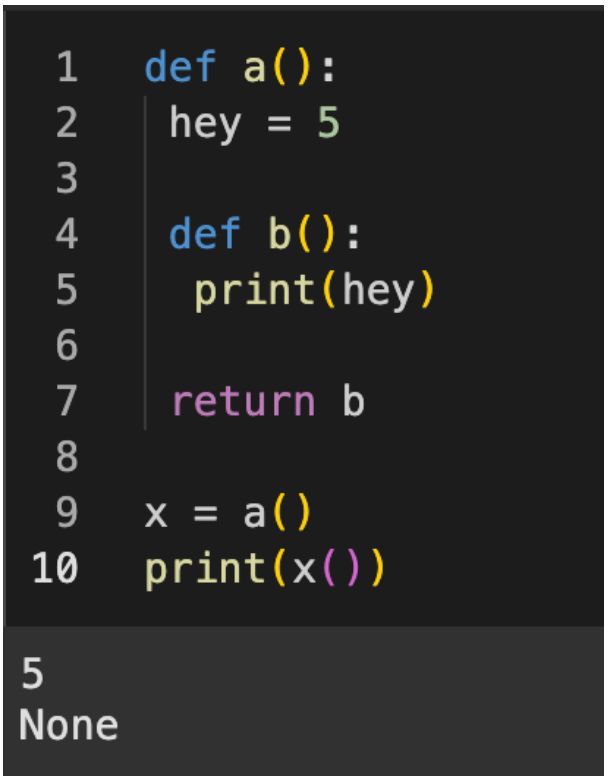
Soal	Keterangan

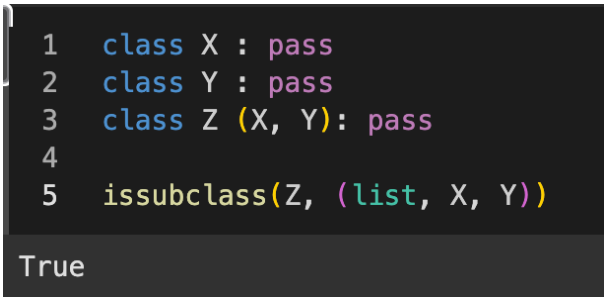
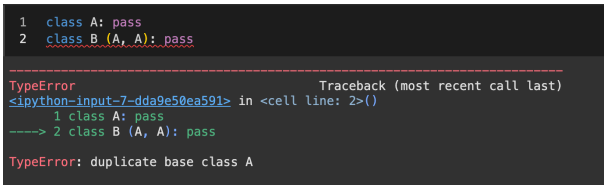
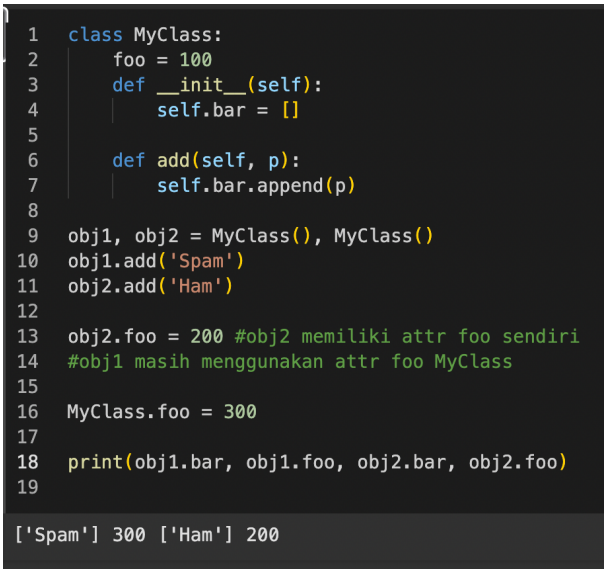
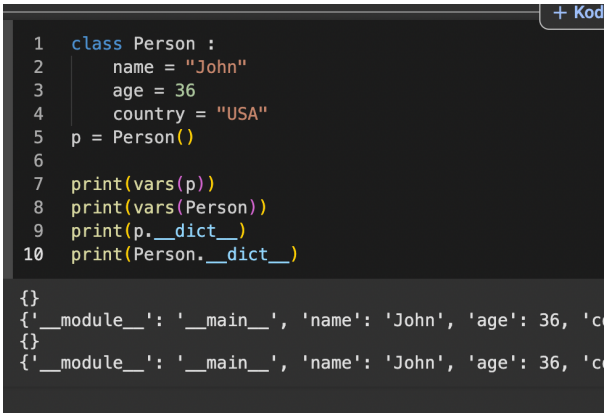
What is the output of the following code?
`class Foo :`
`  bar = 'spam'`
 	
`f1 = Foo()`
`f2 = Foo()`
`f2.bar = 'ham'`
`Foo.bar = 'eggs'`
`print(f1.bar, f2.bar, Foo.bar)`
( ) spam ham eggs
( ) eggs ham eggs
( ) eggs eggs eggs
	Jika instance tidak melakukan rewrite `bar` maka `bar` akan mengambil attr kelas

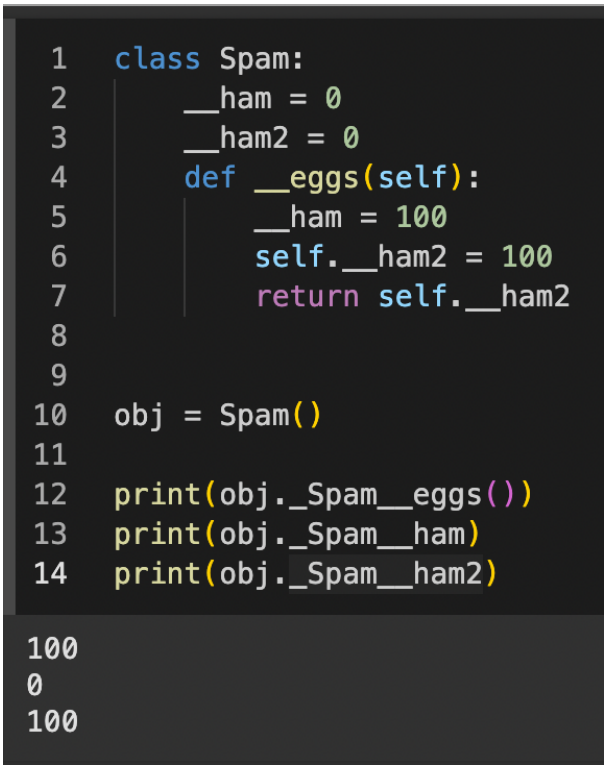
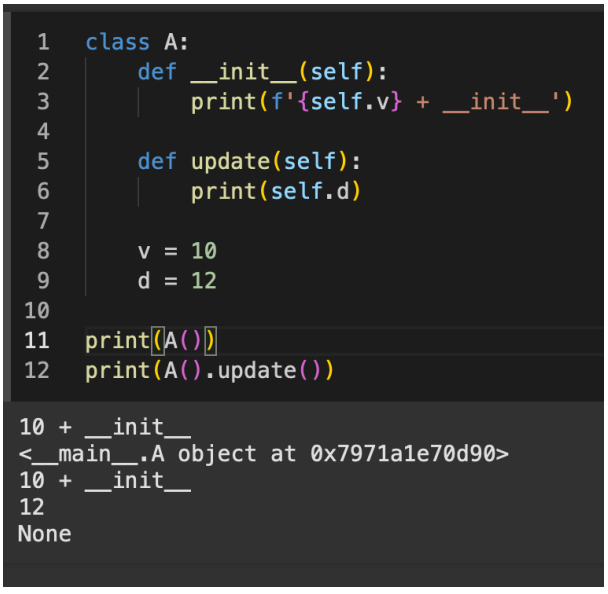
`issubclass(classA, classA)`
will return `True`
 	
`isinheritance(classA, classA)`
will return `True`	

	because when modify string, python will make a different reff memory `is` used to compare 2 object memory
	dua instance pasti memiliki beda referensi memori
JANGAN GUNAKAN `assert` dalam production	dengan perintah `-0` ketika run program di CLI maka semua `assert` akan unactive
	`lambda` bisa di assign ke variable

Soal	Keterangan
	pendekatan terbaik untuk return menggunakan get(), bukan menggunakan index. jika menggunakan index, ketika key tidak ada maka akan error, sedangkan jika dengan get() itu akan mengembalikan nilai default yaitu 0
	A -> kelas A() -> instance
	obj2 merubah value obj1 karena merujuk pada memori yang sama
	A() adalah instance jadi memiliki c Cara baca: `Apakah A() memiliki attr 'c'`

Soal	Keterangan
 	karena tidak ada return maka default mengembalikan None
`isinstance(obj, cls)`	**hanya berlaku di instance (obj)** Mengecek apakah suatu **objek** adalah instance dari suatu kelas atau subclass.
`issubclass(cls1, cls2)`	**hanya berlaku di kelas** Mengecek apakah suatu kelas adalah subclass dari kelas lain.
What is the output of the following code? **class Ham** : **def __init__** (self): self.v1 = 1 **class Spam** (Ham): **def __init__** (self): self.v2 = 2 s = Spam() print(s.v1,s.v2) ( ) 0 2 ( ) 1 2 ( ) Invalid Syntax ( ) AttributeError: 'Spam' object has no attribute 'v1' Answer ~~~	ketika mendeklarasikan `__init__` sendiri maka untuk memunculkan attr pada kelas induk harus menggunakan `super().__init__()` atau `Ham.__init__(self)`

Soal	Keterangan
	`foo` disimpan ke bar tanpa di eksekusi, hanya menyimpan referensi sedangkan jika `foo()` akan error karena ketika menyimpan ke bar, `foo()` akan ter eksekusi
	Apakah `Z` adalah salah satu subclass dari tuple(list, X, Y) - bukan subclass dari `list` - subclass dari `X` - subclass dari `Y`
	can't duplicate
	setiap instance memiliki attr sendiri, ketika tidak mengoverwritte attr kelas maka akan selalu menggunakan value pada attr kelas
	`vars(Class)` dan `Class.__dict__` memberikan output yang sama, hanya pendekatan `vars()` lebih aman

Soal	Keterangan
	`type(self)` -> menghasilkan representasi kelas `.__name__` menghasilkan nama kelas
	Kenapa `obj._Spam__ham` mengakses attr kelas `__ham` alih alih mengambil `__ham` di method `__eggs()` :  Gunakan `self.__ham` agar menjadi attr instance dan bisa di panggil oleh object
 	# `print(A())` - method di dalam kelas bisa mengambil attr walaupun deklarasinya setelah method - `print A()` akan menghasilkan yang ada di `__init__` dan output kelas `<__main__.A object at 0x7971a1e719f0>` # `print(A().update())` - memanggil kontructor `__init__` - memberikan output `self.d` - print method tanpa return secara default memberikan output `None`

Soal

```
class Ham :
    def __init__ (self):
        print(type(self).__name__ + '.__init__()' , end= ' ' )
        self.update()
    def update (self):
        print(type(self).__name__ + '.update()' , end= ' ' )

class Spam (Ham):
    def update (self, param):
        print(type(self).__name__ + '.update(param)' , end= ' ' )

Ham()
Spam()
```

() Ham.__init__() Ham.update() Ham.__init__() Ham.update()
() Ham.__init__() Ham.update() Spam.__init__() Spam.update()
() Ham.__init__() Ham.update() Spam.__init__() Spam.update(param)
() TypeError: update() missing 1 required positional argument: 'param'

Answer >>>

Keterangan

`Spam().update()` membutuhkan argumen 'param' karena `__init__` pada kelas Spam memanggil `update(self, param)` dikelasnya sendiri

---

`Spam().update()` akan mengambil dari kelas Ham ketika `__init__` dijalankan, sehingga tidak membutuhkan argumen param.

Karena `__update = update` dan `self.__update()`

perhatikan bahwa `__update = update` bukan `__update = update()`, karena jika ada kurung `()` method akan tereksekusi

```
class Ham :
    def __init__(self):
        print(type(self).__name__ + '.__init__()' , end= ' ' )
        self.__update()
    def update(self):
        print(type(self).__name__ + '.update()' , end= ' ' )

    __update = update
class Spam (Ham):
    def update (self, param):
        print(type(self).__name__ + '.update(param)' , end= ' ' )

print(Ham())
print(Spam())
```

```
Ham.__init__() Ham.update() <__main__.Ham object at 0x7cd80e533f10>
Spam.__init__() Spam.update() <__main__.Spam object at 0x7cd80e533c40>
```

```
class Spam :
    __ham, ham = '__ham' , 'ham'
    def __eggs (self):
        pass
    eggs = __eggs
s = Spam()
print(Spam.__name__)
# print(s.__name__)
print(s._Spam__eggs.__name__)
# print(s._Spam__ham.__name__)
print(s.eggs.__name__)
# print(s.ham.__name__)
```

```
Spam
__eggs
__eggs
```

`__name__` ; `__module__` ; `__bases__`

- `__name__` : mengembalikan nama

`print(__name__)` -> `__main__`

- `__module__` : mengembalikan module `__main__` atau module lain jika diimport

- `__bases__` : untuk melihat hierarchy pewarisan

Tabel Perbandingan:

Atribut	Dari Kelas	Dari Instance	Cara Akses dari Instance
`__name__`	✓	✗ (akses melalui `__class__`)	`obj.__class__.__name__`
`__module__`	✓	✓	`obj.__module__`
`__bases__`	✓	✗ (akses melalui `__class__`)	`obj.__class__.__bases__`

Penjelasan `type` sebagai pembuat kelas

Fungsi `type` biasanya digunakan untuk mengetahui tipe suatu objek, misalnya:

python Copy code

```
print(type(5))  # Output: <class 'int'>
```

Namun, `type` juga bisa digunakan untuk membuat **kelas** secara dinamis. Formatnya adalah:

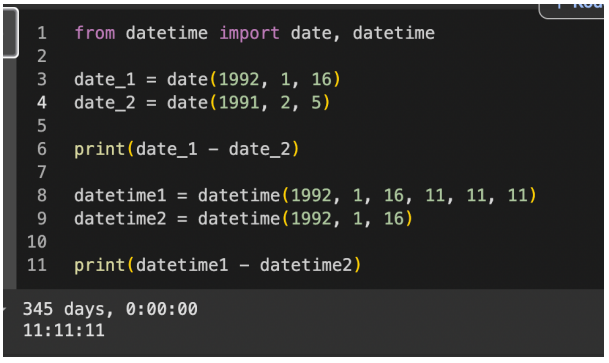
python Copy code

```
type(class_name, bases, attributes)
```

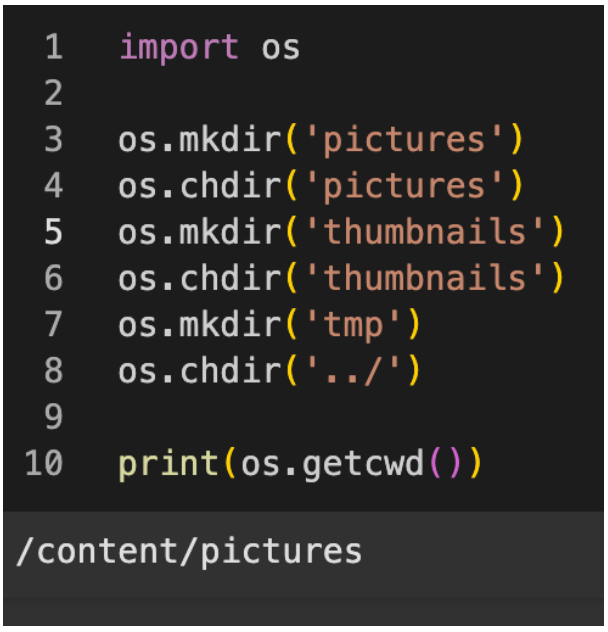
- `class_name` : Nama kelas yang ingin Anda buat.
- `bases` : Tuple berisi kelas-kelas yang menjadi parent (base class).
- `attributes` : Dictionary berisi atribut atau metode untuk kelas.

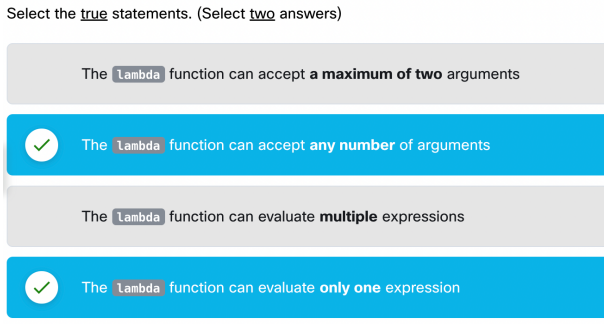
Soal	Keterangan
1 0%0 --- ZeroDivisionError Traceback (most recent call last) <ipython-input-46-5d2de8fe0392> in <cell line: 1>() ----> 1 0%0 ZeroDivisionError: integer division or modulo by zero	mod 0 -> error

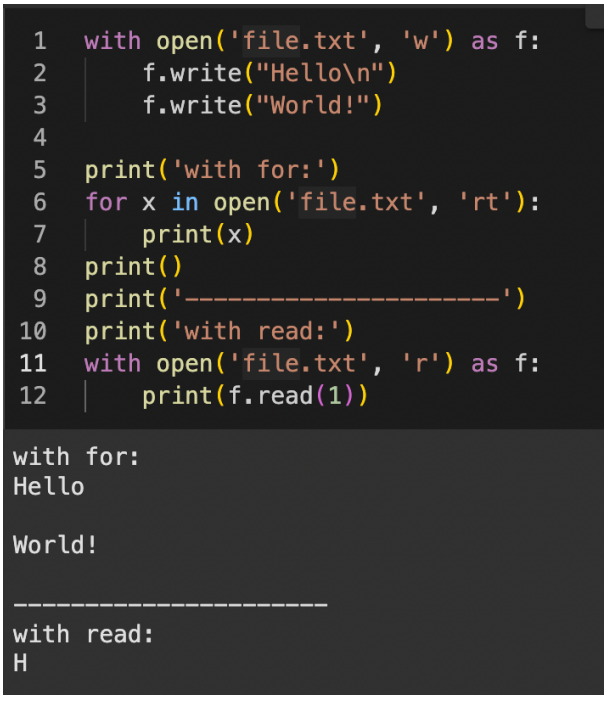
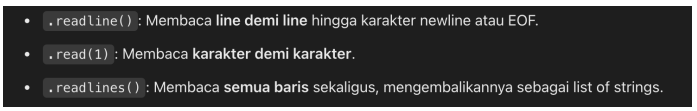
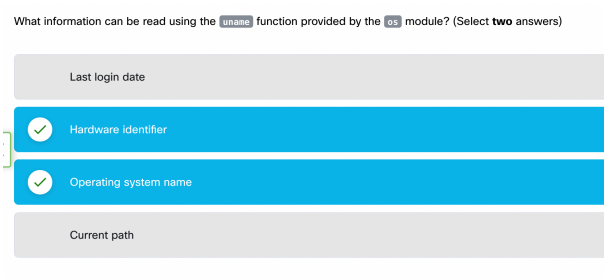
Miscellaneous

Soal	Keterangan
1 import calendar 2 3 c = calendar.Calendar() 4 5 for weekday in c.iterweekdays(): 6 print(weekday, end=" ") 7 8 0 1 2 3 4 5 6	akan menghasilkan dari 0
1 from datetime import date, datetime 2 3 date_1 = date(1992, 1, 16) 4 date_2 = date(1991, 2, 5) 5 6 print(date_1 - date_2) 7 8 datetime1 = datetime(1992, 1, 16, 11, 11, 11) 9 datetime2 = datetime(1991, 2, 5) 10 11 print(datetime1 - datetime2) 345 days, 0:00:00 345 days, 11:11:11	jika hari = 0, maka tidak muncul  1 from datetime import date, datetime 2 3 date_1 = date(1992, 1, 16) 4 date_2 = date(1991, 2, 5) 5 6 print(date_1 - date_2) 7 8 datetime1 = datetime(1992, 1, 16, 11, 11, 11) 9 datetime2 = datetime(1992, 1, 16) 10 11 print(datetime1 - datetime2) 345 days, 0:00:00 11:11:11

1 import os
2
3 os.mkdir('thumbnails')
4 os.chdir('thumbnails')
5
6 sizes = ['small', 'medium', 'large']
7
8 for size in sizes:
9 os.mkdir(size)
10
11 print(os.listdir())
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
(.venv) naufalnashif@MacBook-Air-Naufal python-
/Users/naufalnashif/Desktop/python-dev/main.py
['large', 'medium', 'small']
(.venv) naufalnashif@MacBook-Air-Naufal python-
	bakal diurutkan sesuai abjad
	`['large', 'medium', 'small']`
	dengan urutan abjad :
	`a, b, ..., l, m, ..., s, ..., z`

Soal	Keterangan
	sudah kembali ke path `/pictures` dengan perintah `os.chdir(../)`

	`lambda arguments (bisa banyak): expression (only one)`
	
	`lambda x , y, z : x + y + z`

	menggunakan `for` akan membaca line by line sama seperti `.readline` sedangkan menggunakan `.read` akan membaca karakter demi karakter 
	`os.uname` hanya ada di unix/linux

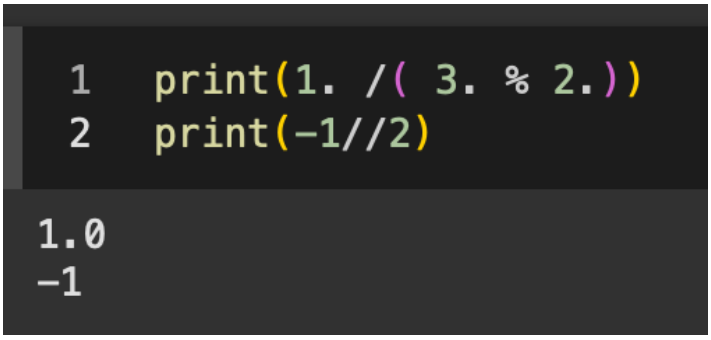
Soal	Keterangan

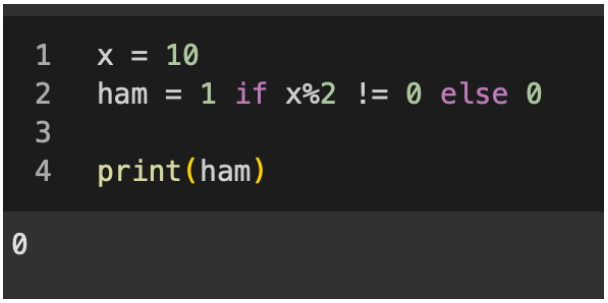
Ya, data biner yang diberikan **(Hexadecimal):** `0x01 0x02 0x03 0x04` terdiri dari **4 byte**.
• **Penjelasan**:
  • Setiap nilai dalam format **hexadecimal** seperti `0x01`, `0x02`, `0x03`, dan `0x04` mewakili **1 byte**.
  • Total ada 4 nilai, sehingga ukurannya adalah **4 byte**.
  • Dalam Python, Anda dapat membuat buffer yang sesuai dengan ukuran ini menggunakan `bytearray(4)`.
	`bytearray(4)` -> membuat array dengan panjang 4 byte

1 x = 10 2 y = 15 3 4 f = lambda i : i + x + y 5 6 f(2) 27	deklarasi lambda bisa seperti ini

python Copy code
`import sys`
`temp = sys.stdout  # Simpan referensi stdout asli`
`sys.stdout = open('spam.txt', 'w')  # Ubah stdout ke file 'spam.txt'`
`print("Hello World")  # Output ini ditulis ke 'spam.txt'`
`sys.stdout.close()  # Tutup file 'spam.txt'`
`sys.stdout = temp  # Kembalikan stdout ke konsol asli`
`print("Good Bye")  # Output ini muncul di konsol`
	`sys.stdout` secara default diarahkan ke konsol, sehingga `print('string')` akan muncul di konsol.
	
	Namun ketika di arahkan ke file `spam.txt` `print('string')` bukan muncul di konsol melainkan di `spam.txt`

What is the output of the following code? a, b = 10 , 20 print(a < b **and** a **or** b) ( ) 10 ( ) 20 ( ) True ( ) Invalid Syntax (True and 10) or b 10 or b -> 10 ———————— and -> 10 ; or -> jika pertama sudah True maka tidak cek argument berikutnya	`True and 10` -> mengembalikan operand ke 2 karena operan pertama `True` Jika `or` , jika operand pertama `True` maka akan langsung di kembalikan
Select the values considered false. [ ] None	
[ ] False [ ] zero of any numeric type (0, 0L, 0.0, 0j) [ ] empty sequence ('', (), []) [ ] class with a __nonzero__() definition [ ] class with __len__() that returns integer 0	
Select valid integer assignment for the variable spam? [ ] spam = 1e0 [ ] spam = 0b1 [ ] spam = 0o1 [ ] spam = 0x1 [ ] spam = \u0031 #The Unicode of 1 is U+0031 Answer >>>	

Soal	Keterangan
	
	-1//2 -> floor(-1 / 2 = -0.5) sehingga -1
Select all keyword argument of print() [x] sep [x] end [x] file [x] flush [ ] format	`print(sep, end, file, flush)`
What is the output of the following code? 1 print(int(10.10), end= " ") 2 print(int("10" , 10), end= " ") 3 print(int("10" , base= 10), end= " ") 4 print(int(0o12), end= " ") 5 print(int(10)) (x) 10 10 10 10 10 ( ) TypeError: int() takes at most 1 argument (2 given) ( ) TypeError: 'base' is an invalid keyword argument for int() ( ) Invalid syntax on Line 4 (0o12)	
What is the output of the code? 1 print(float('+1.23'), end= " ") 2 print(float(' -12345\n'), end= " ") 3 print(float('1e-003'), end= " ") 4 print(float('+1E6'), end= " ") 5 print(float('-Infinity')) (x) 1.23 -12345.0 0.001 1000000.0 -inf	
What is the output of the following code? >>> 5 ** 0 ** 0 ( ) SyntaxError: invalid syntax ( ) 1 (x) 5 ( ) 0	dalam python `0**0 = 1` walaupun dalam matematika `0^0 = 0`

Soal	Keterangan
	penulisan if else bisa seperti itu

---

Thank You !!