

List = [۱, ۲, ۳]

: append *

List.append(۴)

(درسته کنندگان اینجا اضافه کریں کس)

print(List) → [۱, ۲, ۳, ۴]

len *

List = [۱, ۲, ۳]

درسته کنندگان اینجا اضافه کریں کس

len(List) → ۳

یا همان طول بوده

extend *

(برای افزایش کردن بینی در یک آرایه باید از درسته کنندگان استفاده کنید)

List = [۱, ۲, ۳]

برای افزایش اندک را در بردارنده افزایش دادم

List.extend([۴, ۵, ۶])

print(List) → [۱, ۲, ۳, ۴, ۵, ۶]

لذا این درسته باید جنبه ای داشته باشد که این را مستعار کرده باشد

List_1 = [۱, ۲, ۳]

List_2 = [۴, ۵, ۶]

List_1.extend(List_2)

print(List_1) → [۱, ۲, ۳, ۴, ۵, ۶]

* جنبه ای داشته باشد

List_1 = [۱, ۲, ۳]

List_2 = [۴, ۵, ۶]

print(List_1 + List_2) → [۱, ۲, ۳, ۴, ۵, ۶]

* افزایش ها

با این دو پایه ای از همین روش غیر معمولی که ممکن است این را درسته کنندگان استفاده کنند

List = [۱, ۲, ۳, ۴, ۵, ۶, ۷, ۸, ۹]

List[۳] = ۴

اگر این سوم را انتخاب نمی کنیم که

List[۳:۵] = [۴, ۵] → لذا این سوم کا ایس یعنی (پنجمین ایس تا پنجمین ایس) کو حذف کریں

List[۳:۷] = [۴, ۵, ۶, ۷, ۸] → لذا ایس کو حذف کریں

[item₁, item₂, item₃, item₄, item₅]
↓ ↓ ↓ ↓ ↓
: : : : :

لیست میتواند دنیا بزرگتر است که ایتم هایی داشته باشد

index *

برای بررسی آوردن اندیس چهارم از لیست کافیست که index را بخواهیم که این اندیس برابر با ۳ خواهد بود.
List.index("Hello") → ۰
List.index("wife") → ۱

count *

برای تکمیل باین کار باید در لیست همین بارگذاری شود. این کار را count میگویند.
List = [۱, ۲, ۳, ۴, ۵]
List.count(1) → ۲
List.count(۳) → ۱

sort *

لیست های که عمق بالا دارند از این نظر مغایر باشند. مثلاً میتوانند این را sort کنند.
List.sort()
print(List) → [۰, ۱, ۲, ۳, ۴]

عنوان این اقدام این است که لیست میتواند مرتب شود.

List.sort(reverse=True)

print(List) → [۴, ۳, ۲, ۱, ۰]

که معنی این است که این های را سهی بپرسیم، به ترتیب معروف از بزرگ به کوچک.

clear *

List.clear()
print(List) → []

Pop *

برای حذف یک ایتم از لیست pop را میگذرانیم که این ایتم را حذف کاری میکند.
List.pop(1) → ۲
print(List) → [۰, ۳]

`a=1..... b=r..... c=r..... print *`

print("Hello World")

day = 1 month = 9 year = 2010

`print(day, month, year) → A 9 x 9.`

```
print('day'; day, 'month'; month, 'year'; year) → day: A month: B year
```

بـهـدـرـهـ بـالـفـرـقـانـ دـيـلـهـوـرـهـ مـرـسـهـ قـوـهـدـيرـ

درینه ایم ضریب بدغور حفظی عبد العزیز لش استادی کنی:

```
print("day:", day, "\n", "month:", month, "\n", "year:", year)
```

day: A

month: 9

\year := 100;

..... del *

لطفاً حفظ است و میتوان فی الحال مل متفقین با عنصرهای delete

type *

بالاستثناء من those types التي تدخل في نوع المغيرات المعنوية

`type(day) → int`

* علم رفیع بدرها سیل و علیریسم در گیلان / می باشد

خواسته شد و پسوند با نیز ملکه عیا شد

با قوی ماده سرمه بسته با ۱٪ منیزین آلام را بخوبی برداشت می‌کند.

تولیت سرمایه‌گذاری می‌نماید.

print(c) → 50

لقد تم عذرها في المقابلة : $d = a + \alpha b$

- * علیر = + در پایان :
 $sum = 0$
 $sum + = a$ لزماً از این شرط که به متغیرها حق واقع افزاوی داشته باشند :
- $print(sum) \rightarrow v$ $sum = sum + a$ و $sum += a$ هستند
- * علیر = - در پایان :
 $diff = 0$
 $diff -= v$ بجز اگر مقداری از متغیر استفاده نموده باشد .
- $print(diff) \rightarrow v$
- * علیر = * در پایان :
 $mul = 1$ برای هر سه متغیر در متغیرها ابتدا عدد ۱ را داشته باشند .
- $mul *= v$
- $print(mul) \rightarrow 1v$
- * علیر = / در پایان :
 $div = v$ بجز تقسیم عددی در متغیرها ابتدا عدد ۱ را داشته باشند .
- $div /= v$
- $print(div) \rightarrow 1,0$
- * علیر = // در پایان :
 $num = 1$ خارج هست تقسیم دو متغیرها به یک عدد اینست لآن دو هست .
- $num // = v$
- $print(num) \rightarrow v$
- * علیر = % در پایان :
 $rem = v$ با همیک مذکور شده تعبیم متغیر بر علاوه دوستانی دارد .
- $rem \% = v$
- $print(rem) \rightarrow v$
- * علیر = ** در پایان :
 $pow = 1$ بجز اگر متغیرها به یک عددی ابتدا عدد ۱ داشتم :
- $pow ** = v$
- $print(pow) \rightarrow 4^v$

رئیسی

Subject: مبانی برنامه نویسی
Year: Month: Date: ()

name = 'ehaam' * دو قسم (۱) یارابیل کوئی (۲) مزقی باید ذکر شود.

l-name = "hifa" * باین ترتیب رئیسی مطابرست ندارد فراز نام را دون بیل کوئی نیاز ندارم.

* حقیقتی مطابرست دو قسم بلطف پاسخون سه کار عدد صواب بھی نہیں بلکہ آنرا رئیسی میں داند.

* رئیسی دریافتیں در واقع لستی لفظ اکثر میں کوئی مانند آئین جاگی بھی لستی قابلیت

txt = 'I love python' لندسین دھنی طاقت

print(txt[0]) → I عی توکل جا لندسین دھنی هر عبارتی را لندسی سے بیرون کشی

print(txt[7:-1]) → python در پاپرٹ کیلئے کارکتر ہے صواب بھی آئسے

* رئیسی حافظہ پالیٹ قفسہ ناپسیدا immutable

print(len(txt)) → 11 طول رئیسی سے پسیدا

txt1 = 'python' جیسا کہ رئیسی دھنی کوئی

txt2 = 'course' + بار ایک توور

print(txt1 + txt2) → Python.course

print(txt1 + ' ' + txt2) → Python course

* بار اسکو : join :

s = '' در این روئی با استفادہ از کارکری کو فریمان منحصر بھی کیم

print(s.join([txt1, txt2])) من مطلقاً چیزی رئیسی را بھم مسئلہ نہیں

→ Python course این کارکری عنی کوئی کوئی space بھی نہیں کیا ویرگول یا ہر قسمی کوئی بار

s = '*'

print(s.join([txt1, txt2])) → Python*course

* دسکو find

print(txt.find('python')) → ۵ ایک بار اس کیلئے لندسین بیل معتبر نہیں

~~کارکر عبارت میں دسکو ایک خروجی دسکو find ایک خاص بردار~~

`print(tet.find('python'))` → ۷

```
print(text.find('Python')) → -1
```

کلیه این خواص ملکوتی را می‌توان درست نمود که بین دو کلیک بازگشتی می‌باشد از این دو کلیک می‌توان معمولی و غیر معمولی می‌داند.

`txt = I love python` : replace `love` with `python`

برای ...**چاکر** زدن کوتاه میکنند و درسته با عبارت دلخواه بکار روند.
new_txt = txt.replace('I', 'you');

`print(brew_tat) # you love python`

① print(txt.split(' ')) : split

→ ['I', 'love', 'python']

٥ print(txt.split(' ')) لطباعة كل الكلمات من المدخل في مصفوفة متعددة الأبعاد

`→ ['I.I.', 'very', 'n']`

د. مورخ. دوسم طالع فیداللهه دا. د. ناصر مرتضی دفعه هر چارماه ۰ بود. رئیسیت لوزه هنری بررسی خود را امس

..... ﴿۱۰﴾ عَلَيْكُمْ نِسَاءُكُمْ وَرِبَّهُمْ مُؤْلِمٌ

`print(f == 14/f)` → True

`print(v == y)` → False

print('a' == 'A') → False کوچکتر کیا تو ایسا کوچک نہیں

۱۰۰۰ علیرضا مساوی در پایان سال = *

`print(f != 141f) → False` عبارت مساوی دفعه مالص بگذشت سیاری عمل نماید.

لطفی دفتر علم ارایت بسته قمپور را است بفرار می‌باشد

`int(a != 'A')` → True `a = False` `a == False` → True

point (15.7.v.) \rightarrow True : **Cause** بیرونیتکی درمانیوں کے \star

`point(v > 1r) -> False`


```
a = r
b = r
if a > b:
    print('a is greater than b')
elif a < b:
    print('a is less than b')
else:
    print('a is equal to b')
```

for var in iterator: (for نہاد) for دستور *

loop body مجموعه ای از ریتارڈ کے مکالمہ است و یعنی

در صورت تغیراتی از ترتیب حالتی داخل است. پس از آنکه سی اند و مودعی

متغیر var بین تعدادی متغیر loop body فرماتگار گونه

تعریف شده است.

list = [1, 2, 3] مطالعه ای که در ضمیمه نمایش داده شوند ←

for num in list:
 print(num) : for و if و *

list = [1, 2, 3, 4, 5] مجموع فرماتگاری ایجاد کنید

for num in list:

if num % 2 == 1:
 print(num) ← ۱

print(num) ← ۰

while condition: : while نہاد *

loop body عباراتی که while کے مطابق ایجاد کرده است.

condition بوقتی که loop body کی ایجاد کردن از دستور است.

cnt = 1 (مطالعه کرده که شرط همراه برقرار رسانید. آن خاصیت که ایجاد کردن ایجاد کردن)

while cnt <= 0: (مطالعه کرد که for و while دستوراتی که کنند)

print(cnt) (برای این کار که در نظر داشتیم که کسی نیز نداشتیم. بعد این کار رسم کنید)

cnt += 1 (مطالعه کرد که در نظر داشتیم که کسی نیز نداشتیم. با این دستور دستوراتی که کنند)

→ ← ۱ ۲ ۳ ۴ ۵

number = 1 : while * مثل لز جلت
 while number <= 100 : {
 print(number) }
 number += 1 طبق

* می سبب فاکتوریل با صاف؟ : while
 num = 1 خاکسوزی عدد برابر با 1 می باشد طبق دو پایه از قدر است
 int = num - 1 پس به بعدی عاشر و ملة while بیار درهم
 while int > 0 : علیعه که فاکتوریل را برای آن می سازی کمین num و int و
 num *= int int خان را دریند سه کم می کند تا باقی کسر از num باشد
 int -= 1 int = 1
 print(num) طبق ۲۴

def function-name(args) : کابیچ
 [] statements
 return value (اطیروزه return دخواه ایسته می باید کامبرد یعنی بازیابی شود)
 def حسنه که ناتایج با این دلیل هستند می خواهیم
 args ناتایج و درودی های ایسته ناتایج باید داشته باشند. همان ایسته هایی که بخوبی درودی خواهند داشتند و درین می خواهد
 args بازیست اما ایسته کامبرد هایی داشتند که بخوبی درودی خواهند داشتند و درین می خواهد
 درهم خواهد داشت
 statement هواسته هایی که از ایسته داریں حست خواسته شوند
 return در محدودیت که بخواهیم ناتایج خواهی را نه با ایسته بازیابی کرد return باید برگرداند ضروری است محدودیت نداشته باشند
 در این صورت بعده ای هیچ ایسته که از ایسته داریم خواهد داشت و return (دو یا یکی value) خواهد داشت

* برای ایسته دلخواهی ای ایسته که ایسته داریم و مطوبیت بدل پرانتز باز کشید و درودی بیضیم عالی
 print(sum-list) ۹ طبق
 دلخواه ایست برویم فرمول بخواهیم داشت ایسته داریم و return را بخواهیم داشت

دیکشنری در پایتون \rightarrow دیکشنری داده (data type) برای ذخیره داده است. دیکشنری بجزءی از داده ها که لست است. دیکشنری داده های ذخیره شده با جزویتی که در مردم رایم که ذخیره می شوند \leftarrow همان ترتیب با لغزش دهن. مکانیزم کلی دیکشنری را برایم.

ضروری

* داده های دیکشنری \rightarrow key:value \checkmark دیکشنری ذخیره می شود. هر قسم دیکشنری key:value \checkmark داشت. می کند \rightarrow دیکشنری ماتریس خوبی نیست. هر قسم دیکشنری دیکشنری دیگر. داده های آن دیکشنری ذخیره نمی شود. ماتریسیست حقیقتی کوچک نباشد \rightarrow دیکشنری داریست.

* برای ذخیره دیکشنری در پایتون لز اول $\{ \}$ استفاده می شود. برای داده های دیکشنری key:value \rightarrow دافلی $\{ \text{key} : \text{value} \}$. و قابل اطمینان هست. می توان می دیکشنری ذخیره نمود. my_dictionary = {key1: value1, key2: value2, key3: value3, ...}

(countries) می خواهیم پایه داشت دیکشنری ایران و انگلستان را ذخیره. دیکشنری ذخیره داشت و خوبی داشت. countries = {
'Iran': 'Tehran',
'England': 'London'}

print(countries) \rightarrow خوبی $\{ 'Iran': 'Tehran', 'England': 'London' \}$

(grades) ۲. نمره لزه را داشت. بوز را در دیکشنری ذخیره. بعنوان: grades. دیکشنری ذخیره داشت. grades = {
'kate' : [80, 88, 100, 92],
'jane' : [70, 89, 85, 90]}

print(grades) \rightarrow خوبی $\{ 'kate' : [80, 88, 100, 92], 'jane' : [70, 89, 85, 90] \}$

اول فقرہ در درجہ دیکشنری key: value

برای این کار کافی است بنویم dict-name.(key)=value

(Q) بی خواهیم پایتخت هر کشور را در دیکشنری countries ذکر نمودیم:

countries['India'] = 'Delhi'

print(countries) → { 'Iran': 'Tehran', 'England': 'London', 'India': 'Delhi' }

* استخراج یک مقدار از دیکشنری:

برای استخراج یک مقدار از دیکشنری باید کلید را از بین کلیدها برداشته باشد.

* دینه مقدار key را عنوان می‌کنیم و مقدار value را نیز

(Q) بی خواهیم نوشته Jane را در دیکشنری grades بایسیم:

Jane-grades = grades['Jane']

print(jane-grades) → [۹۰ و ۸۹ و ۵۵ و ۷۸]

* این دلیل دیکشنری را می‌توان بنا کرد که (Marry (ولد کنیم) با خطا مواجه نمی‌شوند

* تفسیر value در دیکشنری

مثال دلیل grades را در دیکشنری کار کنیم:

grades['Kate'] = [۷۰ و ۸۸ و ۵۰ و ۹۰]

print(grades) →

با این کار نتیجه قبیح وارد شده در مفروضات خوبی نمی‌گذرد.

* تفسیر key در دیکشنری به عنوان سمت اعلان می‌شود

★ بحثی و جویدن مقدم و مجدد ملک key در دیکشنری :

اگر تابعی را فراخوانی کنیم که در دیکشنری وجود نداشته باشد، باعدها مواجه قواعدی خواهد شد.

پس قبل از فراخوانی آن کلید، ابتدا این میررسی می کنیم که آن key در دیکشنری وجود دارد یا خیر؛
برای این کار لازم کلید و لذت in استفاده می کنیم

`print('jack' in grades) → False`

`print('jane' in grades) → True`

★ لیست کردن key های یک دیکشنری در یک سو :

برای استرجاع key های موجود در دیکشنری keys، لزوماً grades را استفاده می کنیم:
`grades.keys = grades.keys()`

`print(grades.keys) → dict_keys(['jane', 'kate'])`

`print(type(grades.keys)) → <class 'dict_keys'>`

کاربرون با متفقینی از نوع dict-key کسانی هستند. می توانیم کتابچه متنی ملک لیست تبدیل کنیم:

`list_keys = list(grades.keys)`

`print(type(list_keys)) → <class 'list'>`

★ لیست کردن value های یک دیکشنری در یک سو :

برای لیست کارکاری لست کردن values() استفاده می کنیم مثلاً:

`grades.values = grades.values()`

`print(grades.values) → dict_values([{'name': 'jane', 'grade': 90}, {'name': 'kate', 'grade': 89}, {'name': 'mike', 'grade': 95}])`

فیکس کردن مقداری خواهی فرمود که در اینجا آن کاملاً تراست.
`list.values = list(grades.values)`
`print(list.values) → [[80, 85, 90, 90, 90], [85, 89, 90, 90, 90]]`

جذف: مدارس لرن... (لیکنسری)

حروف Keys:Value لزدیسترنی :

برای اینکار لز دستور `pop` است که کامپیوٹر `key` را از `dict` حذف کند. `key` و `value` مطابق با `pop` هستند.
`grades.pop('kate')`
`print(grades)` → `{'jane': [98, 99, 100]}`

۴- حذف همای ملنا میر دلخیزی

لزورستور clear استفاده می کنم :
grades... clear () دیکشنری را بروز نگیریم .
print(grades) → [] لیست به جویی Value هارا نمایش دهیم با . (کسیم جبرید افتخاف کنیم با ...)

عَلِيٌّ الْمُسْكُوْنِي

با این روش `f-string` باید میتوان `print` را با داده در واقع میم رسانده است و حقیقتی
قابلیت های بیشتری نیز دارد. مثلاً:

- ۱- می توان دافل `f-string` هفچتار کسر میانه دافل `f-string` را بازگردانید. مثلاً:

```
def greeting(name):
    return f'Hello {name}'
```

- print(greeting('zahra')) → Hello zahra

```
def greeting(first_name, last_name):
```

```
return p'Hello {first-name} {last-name}'
```

```
print(greeting('Zahra', 'Ramezanroudi')) → Hello Zahra Ramezanroudi
```

..... بینشید که این **interpolated string** یا **f-string** است.

کتابخانه **math** را از داده بنماید:

سلاسل کاراکتر را با وظایفی به صورت **import math** بروزرسانید.

import math

print(f' PI : {math.pi}') → PI: ۳,۱۴۱۵

حال می خواهیم **PI** که عددی تا حدود رقم بعد از اعشار را از داده بگیرد.

برای این کار از **round** استفاده کنید. (بعنوانی بفرز اعشار را بعنوان عددی در داده بگیرد)

print(f' PI : {math.pi:.1ff}') → PI: ۳,۱۴۹

برای این کار از **float** استفاده کنید.

'hello + name' : مرضی من کهنه چیز greeting را درین فرم پرینت کنیم —
 (f') پری ایندر من تولی لز f-string استفاده کرد... بگوئیم که قبل از تو شیوه که این نویسی
 پایین "f" را به عبارت f-string این نام دارد که سری قابلیت های دارد. ("f" در واقع میان رشته است و
 پس از این کاریت حذف شد) این توکل f-string را داریم که این دستور را در این دستور
 این توکل داشت... f-string دستور فرادرد ← دستور f-string و متن میان دو اندیشه ای داشت

def greeting(name):

return f'Hello {name}'

print(greeting('Zahra')) → Hello Zahra

def greeting(first_name, last_name):

return f'Hello {first_name} {last_name}'

print(greeting('Zahra', 'Ramezanrazi')) → Hello Zahra Ramezanrazi

این که این f-string را در اینجا می بینیم این f-string است

۱- بگویی بود که در انتخابات برای انتخاب کنندگان که بزرگتر از ۱۸ سال بودند و در مردم می میان راند

که بزرگتر از ۱۸ سال داشتند اینها را تصویر می نویسیم که اینها کدام انتخاب را داشتند

yes-votes = ۱۳۲۳۴۰۰
no-votes = ۰۱۰۹۸۱۲۳

print(f'{yes-votes / (yes-votes+no-votes)}%')

۲- در این انتخابات که بزرگتر از ۱۸ سال داشتند و در مردم می میان راند

ارائے اگر ملا..... ۱۳۲۵۔ در خود بود)۔ یوں... قاتل سب نہیں کی بھروسے ۱۳۲۶۔ بود)۔ مایکل برلن

بلدرم لیورسٹ ۳۱۲٪ میں ڈاکٹریل ملے۔

نحوه مسند است. در `docs/python` همچنان که می‌توانید بخواهید.

لکھنؤ میری، اپر لیئے فولنائی۔ لسلوڈ مزروں، جن بڑاں... لشکر اسیکا دہ کر دیں۔

~~11818400~~ → 1-~~888~~-400

۲۰. قصیری در محدودیتی (آن لند بقیه) باشد. خوانایی هم برداشته شود.

sum. Y = 1Y

۲۳- ضروری ها را سرتا جهیز سرویس برخواهید ممکن باشد از پایانی میتوان

num. 1 = 15.0

print(num1)..... print(num2)..... print(name) → Sir

```
print(f'{num1:=.2f}')
```

مخرجات المدخلات

```
print(f' {numc:=rd7}')
```

۵۰ شیخ احمد حنفیه در بیان فکر ائمه با برخیری تدوین کرده است. زیرا شیخ حنفیه

```
print(f'{num1+2*cd}')
```

print(f'num1 is {x} and {y}')

point $(f'(x), \text{num}_r^{x=0} \cdot \text{vd}_r f')$

...1,14.....^s~~rule~~

نگارش های برای استفاده از کتابخانه های پاپیون، مژاد سور Import استفاده نمی کنند.

if string contains

```
import datetime
```

```
today = datetime.datetime.today()
```

```
print('f today : %b %d , %Y')
```

.....docs.....April.....5 Apr.....(رقم کتاب).....لارس اولسن

such a dose to be safe.

۲. تابل ۱- تابل دلاری

* قبیط این دو تابه ساخت رخا سه دله ساخت رخای مرکب هستند همین لیست داده دلخواه مرکب است. یعنی کل داده ای که من خواست سابل دلخواه خای دلخواه بود.

تابل:

تابل فیل ضمیمه لیست است. به عبارت دیگر تفاوت بین خواهد باشد.

* برای لیست لز [] استفاده می کردیم. اما برای تابل لزیست انتقال استفاده می کنیم:
tuple = (....)

* در تابل هم لیست هم بولن عدد صحیح هم عدد اعشاری و حجم رئیسه را بدلی خواسته باشیم.

فرق تابل و لیست ← تابل immutable ایست.

(immutale) یعنی نمی توانیم تغییر لیست را در جاگذاری آوریم. تغییر دادن بولن این اتفاق نمی تواند تابل (تابل و جزو متغیر) باشد.

لذا کوچیم تابل را تغییر دهیم باید یک تابل غیر قابل تغییر کنیم → مثلاً بعنوان غیر قابل t = (۳، ۲، ۱) . print(id(t))

این دو تابل (۱، ۰، ۲، ۳) مقداری بعنوان کنیم نیست. print(id(t))

(نکته) اگر برای تابل بر انتقال تابعی به فناوری فرم → بقیه در ۳ و ۲ و ۱ = t داشته باشیم کن کارهای tuple را type(t) می نویسیم. دهن.

پس از کن کارهای داده داده با کامپیوتر این تابل بعنوان داده داده باشد. return (این عبارت که صدیقی دارد)

t = (۱، ۲) * در تابل تهی می باشد. باید بر انتقال بلندیم و زیر قطعی فوراً می شود. (۱۲)

بر این سه برازی آنکه میں استفاده از تابل بقدر فرم → درجه مولود. بجز خروجی دایم از انتقالی کی نظرم

.....پس بگرد و تاکه همه چیز خوبی تایم چیز.....(return).....لز بول تقریباً سه ساعت...وی کلینیک کار فارمی نهادن.....).

زیای از زنده ساخته شد. تاپل استفاده می کنیم. من داریم. چوکر بسته است. تفسیر بدهیم (جاهیت تاپل immutable). لازم است

```

tuple unpacking record record = ('Iran', 1975, 40000000)
country, year, population = record #tuple unpacking

```

☆ می توانیم یه جایی که میخواستیم بمقابل چندین باری چیزی که داشتیم... و متغیر (نام...) را = داده باشیم اما اینها را با $n=12$ یعنی کنم) از $\text{کردن طبق استفاده کرد} \rightarrow \text{عمل}$) $g = \emptyset$ $\left\{ \begin{array}{l} g_1, g_2, \dots, g_n \\ g_1, g_2, \dots, g_n \end{array} \right.$ \rightarrow $\text{یعنی اینجا از triple unpacking} \rightarrow$

* هر کسی اینجا می‌باشد... داده در یکی می‌باشد... داده اینها را که بخواهد... همان‌جا باشند... سعی برای تغییر کردن
دو شخصیت... لز... unpack... کردن... اینست که می‌گذشم... دیگر عورتی نمی‌باشد... همان‌جا را به تغییر دلیری... می‌خواهم...
f-person = 'Alireza' , V.I. + V.A. + 'male' ! میان)

first name) → Alireza ^{میر} وی کسی دیگر را ندارد.

`print(athens) → [+Vg, IN]` ... and all its Leiters pass @ Ein

print(gender) → male اسے لکھنے کا ریکارڈ میں ہاندر اور **کل سائٹ پر** (کالیں سائیٹ) **others** میں رکھ دیں۔

آخرین تقدیر دکوه را صرفهایی می‌خواهیم. من توانم بگویم (منها به عبارت... others... می‌توانیم بگوییم... (ghali...).

وَالْأَنْوَارِ مِنْ تَرَاثِنِي (سَمْ — *others بِلَادِنِي)

﴿وَمَنْعِلُهُ مَرْبُوعٌ وَمَنْعِلُهُ مَدْبُوعٌ بِهِنَا هُنَّ أَهْمَى زَيَادَى مَرْكُومًا نَذَارَهُ . (أوَلَيْكُمْ بِمَدْبُوعٍ هُنَّا هُنَّ أَهْمَى زَيَادَى مَرْكُومًا نَذَارَهُ .) نُوبَانِ (نَوْسَهُ)

* سورهای سیری سیستم استفاده می‌کند. رایج ترین سیری، بیتل (Biel) یعنی اسناد مرتب، مول (Moul) و کانت (Count) و سوپر (Super).

person.[0:2] → ('Alireza' و 'V.') حقوق کا پہلی و ملکی مستہب می توانند میں نظری کرے۔

☆ فرقہ میں ناپل بائیسیں ۔۔۔ ناپل گلابی، اسے دو قابیتیں..... عرضہ میرزا جنگلہر

لے در واقع فایر ایچ جی immutability بدن اسے

* مکاره (۱) چون پیش از این نویسید که چه کنم. این بسته به این است که می خواهد چه کنم. اگرچه همچنان که می خواهد این را از میان برداشته باشد. از این جهت که چیزی که می خواهد از میان برداشته باشد. فرض کنید این می خواهد از میان برداشته باشد. for item in person:
print(item) → {
 'Alireza'
 'Ivr'
 'Xn'
 'mate'

* دستور enumerate: چه کنم؟
for item in enumerate(person):
print(item) → {
 (0, 'Alireza')
 (1, 'Ivr')
 (2, 'Xn')
 (3, 'mate')

در این حالت اگر بخواهد index را بخواهد باشد:

index = item

حال اگر بخواهد کدامیک این سه زیر چیز را بخواهد درست:

دستور for در این حالت اگر بخواهد item را بخواهد باشد:

item = person[index]

برای میان فاصله (بین اندیسها) بخواهد از range بخواهد است:

range(0, len(person))

برای میان فاصله (بین اندیسها) بخواهد از enumerate باشد:

enumerate(person)

برای میان فاصله (بین اندیسها) بخواهد از range باشد:

range(0, len(person))

برای میان فاصله (بین اندیسها) بخواهد از enumerate باشد:

enumerate(person)

برای میان فاصله (بین اندیسها) بخواهد از range باشد:

range(0, len(person))

برای میان فاصله (بین اندیسها) بخواهد از enumerate باشد:

enumerate(person)

برای میان فاصله (بین اندیسها) بخواهد از range باشد:

range(0, len(person))

برای میان فاصله (بین اندیسها) بخواهد از enumerate باشد:

enumerate(person)

برای میان فاصله (بین اندیسها) بخواهد از range باشد:

range(0, len(person))

برای میان فاصله (بین اندیسها) بخواهد از enumerate باشد:

enumerate(person)

برای میان فاصله (بین اندیسها) بخواهد از range باشد:

range(0, len(person))

برای میان فاصله (بین اندیسها) بخواهد از enumerate باشد:

enumerate(person)

برای میان فاصله (بین اندیسها) بخواهد از range باشد:

range(0, len(person))

برای میان فاصله (بین اندیسها) بخواهد از enumerate باشد:

enumerate(person)

list = [۴, ۳, ۲, ۱, ۰, ۲, ۳, ۴] (رده های (List))
def remove_duplicates(li): (برای این روش از ترتیب مکالمه برای این sort کرد)
result = [] (نحوی اینجا فیلتری می شود در راه حل اول نیست)
if not li:
return [] (با اینجا باید باشد)
li.sort()
info = li[0]
result.append(info)
for item in li:
if item != info:
result.append(item)
info = item
return result

def remove_duplicates(l): (لیست را از مجموعه جدا کنید)
if not l: (اگر لیست خالی باشد نیست)
return [] (visited: پیش از این کارهایی که باز است)
visited = {} (آن کاری که دوباره نیست)
for item in l:
if item not in visited:
visited[item] = true (آن کاری که دوباره نیست)
return list(visited.keys()) (این مجموعه را که دوباره نیست)
print(remove_duplicates([۰, ۱, ۰, ۲, ۳, ۲, ۱])) → [۰, ۱, ۲, ۳]

جلسه ۳

حروف صاف و حرف نصف صاف: ۱- لام ۲- همایل ۳- سیمه

حروف صاف و حرف نصف صاف: ۱- لام ۲- همایل ۳- سیمه
حروف صاف: set
حروف نصف صاف: dictionary
حروف صاف: key: value
حروف نصف صاف: list

name = ['Aliceza', 'Mohsen', 'Sarah']
print(type(name)) → <class 'list'>

name = ['Aliceza', 'Mohsen', 'Sarah', 'Aliceza']

print(name) → ['Aliceza', 'Mohsen', 'Sarah', 'Aliceza']

print(len(name)) → ۴

الو جواضیم میل دستور دادوی: set کیم، بسیار کمیز (کمیز) از set میباشد.

کمیز بتوان اینستاکت کیم راجح نموده... فردا در دستور add استفاده فرمیکنیم:

name.add('nazarin')

میتوانیم میل دستور داده... بروی میل دستور داشت باشیم + لز. طبق مفهوم set استفاده فرمی کنیم

پایه معامل: مجموعه (گروهی). اسرع

سؤال: می خواهیم اسامی میتوانند بین فرانگوواری و انگلیسی رو برسانیم؟

برای این مقصد لزیج دستور دادی: ابتدا این دستور میزنیم (۱۶۶ دستور)

print(fa-names.intersection(en-names))

خوبی: {'Sara'}

برای یافتن تفاوت بین دو set باید بین لز. fa-names و دستور difference

لز. برای معرفی این دستور باید مخفی دستور داشت

print(fa-names.difference(en-names)) → {'Mohsen', 'Aliceza'}

لذ. دستور داشت با مجموعه دو کم را به مادر ارساء

* رویی سه مجموعه حسینی بر تولن (۱) کار

print('Ali' in names) → False جو ملکہ تم بیویوں نہ

```
.print('Mohsen' in names) → True
```

للمزيد من المعلومات يرجى زيارة الموقع الإلكتروني www.set.gov.sa

for item in name:

فیلمی جائزہ دریافت کرنے والے (set) کا انتساب point(item) کو سارے Alireza Mohsen کو دیا گی۔

سید علی بن ابی طالب (علیه السلام) کی درود و سوہنگہ کی استاد مدرس

```
stacks = {'Frontend': {'Karfag': 10, 'Madir': 'Frontend'}} (J)
```

```
print(stacks[paatal]) → {'Kangar': 3, 'Modir': 'Felani'}
```

خروجی دلنشی فکر و سکھا

stock = set() مجموعه set ، stock = {} لیستی است.

لیست ملکیت این دستور

سازمان اسناد و کتابخانه ملی جمهوری اسلامی ایران

لستم. حاصلت خاصیت را این‌گونه نیز نداشتم. مثلاً هر دو سرمه‌ای داشتم.

جیون، عصمه، ۱۴۰۰، نهادیا مسکن، ۱۴۰۰، رعنه، بنادرین که کی زنم

```
.....return 1.....
```

.....return ...fib(n-1)+fib(n-r).....

`print(lib.v.)` → x1

$$P(b(n)) = P(b(n-1)) + P(b(n-r)) \quad \forall r \in \{k-1\}$$

سکونتگاهی از این دهکده هایی است که در اینجا میتوانید بخوبی بازدید کنید.

مکالمہ نور میری۔ میری بڑوں میں ایک جسم، بھی عورتی تھی۔

مکالمہ علی الحسن بن علی و علی بن ابی طالب (علیهم السلام) میں مذکور ہے کہ

نامه) اگر بخواهیم مذکور بیشتر خوان (اصحایاری). به دلیل آنکه مسائل در حقیقت کاملاً هستند از مذکور نباید باشند.

۲۰۱۴ **تیر** **۱۳** **شنبه** **۱۴:۰۰** **ساعت** **۰۰:۰۰** **تاریخ** **۱۴۰۳** **تیر** **۱۳** **سال** **۱۴:۰۰** **تاریخ** **۱۴۰۳** **تیر** **۱۳** **سال**

```
def info(name, age=18, city='Tehran')
```

def info(name, age=24, grade):
 print(f'Name: {name}. Age: {age}. Grade: {grade}')

```
printInfo('Alireza', 28) ..... مادر ۲۸ نام: علی‌رضا و روایت آن کاری را در این مقطع بخواهیم
```

اسے یا متعلقہ grade ہے اسی کا لفظ وہ بینگر (bulge) تاریخی ہے اور بدلائیں گے۔ یعنی مفرکر سینے پڑوں۔

لکھوں والے سماں میں اسی طرح

پر اسٹر بلڈ و سبہ عزیز خودی کی کمیت۔ ملاد جوئی راجح فرمے۔ (۱) مضمون کی تکمیل کرنے کے لئے

(Indicates string)

در فروضی دلاری که در وردی ها را به صوره keyword نمایش می‌دهیم، می‌توانیم مثلاً این کلمات را که در متن مورد استفاده قرار گرفته باشند را با استفاده از این کد پیدا کنیم.

```
print(info(city='Tehran', name='Alireza'))
```

Pseudotrichius kennedyi sp. n. (♂)

Keyword city = 'Tehran' Keyword age = ? AND Keyword

..... \leftarrow none \rightarrow للمزيد (طبع).

.....

..... $sl = ('Alireza', 15, 'Tehran')$ تابع sl را تعریف کنید:

print.info(s1))

برای نموده بودم همان خود را

ج پرکی لستہ دلخواہ کا پلے۔ مایہ بی بھروسے فوروم۔ مل مہرید۔ اسے بدی (یعنی ووٹن)، فوٹا نا بی خود ج آن اسے سے

(نحوه) فرخان کارکرد کلی استفاده کردن از میدان خوانای است:
کاربر (متن اسکرین کلی ورودی باشد و باید متن دستور یا کمتر از چند کارکرد را در یک سطر از کنسول کنند) print.info(*S1) tuple unpacking
متوجه شد که این کارکرد متن اسکرین کلی ورودی باشد و باید متن دستور یا کمتر از چند کارکرد را در یک سطر از کنسول کنند) از دروغ این دعاوی tuple unpacking keyword

S1 = {'name': 'Alireza', 'age': 28, 'city': 'Tehran'} (دوست)
print.info(S1['name'], S1['age'], S1['city'])

کاربر نیز صورت مطابق با آن بود

* من توانم مثل روش نویسید این روش را نیز مشاهده نمایم
متوجه شد dictionary unpacking استفاده کنیم
print.info(*S1) را با دوست (**) نمایم چنانچه



نحوه) جی هفاظت به متغیر ورودی نهاده است اما میتوان مفهوم کارکرد کلی و بسط این مفهوم را پژوهش کنیم:

```
def info(name, grades):  
    sum_ = sum(grades)  
    return f' name={name}, avg={(sum_ / len(grades))}'  
print.info('Alireza', [19, 19, 20, 20]) → name=Alireza, avg=19.5
```

نحوه) جی هفاظت به متغیر نهاده است اما میتوان مفهوم کارکرد کلی و بسط این مفهوم را پژوهش کنیم
برای این کارکرد معرفه اولیه است که این فرآیند به صورت (نحوه) ورودی میباشد
def info(name, *grades) grades را فرآیند نماییم
print.info('Ali', 19, 20, 20) tuple unpacking باعث کنند

* tuple unpacking
* name.grades = ('Alireza', 9, 10, 12)
* tuple packing (product, *args)

```

def info(name, grades):
    print(f'{name} درس هایی که دارد: {grades}')
    print(f'{name} درجه هایی که دارد: {grades[0]}, {grades[1]}, {grades[2]}, {grades[3]}, {grades[4]}')

```

```

def info(*grades, name):
    print(info('Alireza', 1, 2, 3, 4))

```

در اینجا آن روش را باید معرفی کرد
با این ترتیب در ورودی (البرای) مقادیر مسکاره داشتم، و باعث شد که

```

def info(name, *grades):
    print(grades)
    print(info('Alireza', 1, 2, 3, 4))

```

میتوانیم این فرم را بخواهیم

مثالی که پس از `*args` میتوانیم چنین اتفاق نماید در عکسی همانجا بدهیم. اگر میخواهیم فرمت `def info(name, *args, city):` را درست کنیم:

```
{ def info(name, *args, city):
    print(info('Alireza', *args, city='Tehran')) }
```

برای این کار میتوانیم از `city='Tehran'` استفاده کنیم و این کار را میتوانیم با `city='Tehran'` نوشت.

حالا اگر میخواهیم فرمت `def info(name, age, *args, city):` را درست کنیم:

```
{ def info(name, age, *args, city):
    print(name, age, *args, city) }
```

برای این کار میتوانیم از `name, age, *args, city` استفاده کنیم و این کار را میتوانیم با `name='Alireza', age=18, city='Tehran'` نوشت.

میرجعیت از پارامترها در فرمت **کوارگز** (kwargs) می باشد. کوارگز معمولاً مقدار پارامترها را در فرمت دیکشنری (dict) می نمایند.

```
def info(name, age, *args, **kwargs):  
    return f' name={name}, age={age}, kwargs={kwargs}'  
print(info('Alireza', 25, city='Tehran'))  
# خروجی: name=Alireza, age=25, kwargs={'city': 'Tehran'}
```

```
def info(name, age, *args, **kwargs):  
    return f' name={name}, age={age}, args={args}, kwargs={kwargs}'  
print(info('Alireza', 25, city='Tehran'))  
# خروجی: name=Alireza, age=25, args=(), kwargs={'city': 'Tehran'}
```

برای اینکه مقدار پارامترها را به صورت ترکیبی (tuple unpacking) در فرمت دیکشنری (dict) در فرمت **کوارگز** (kwargs) بخواهیم، باید از **tuple unpacking** (tuple unpacking) استفاده کنیم. این اتفاق باید با کلمه **args** (args) و **kwargs** (kwargs) که از کلمات کلیدی (keyword) هستند، اتفاق نماید.

```
def info(*args, **kwargs):  
    return f' name={args[0]}, country={args[1]}, age={args[2]}, kwargs={kwargs}'  
print(info('Alireza', 'Iran', 25, city='Tehran', name='Aghamondi'))  
# خروجی: name=Alireza, country=Iran, age=25, kwargs={'city': 'Tehran', 'name': 'Aghamondi'}
```

```
def power(a, b):  
    return a ** b
```

مقدار پارامترها را می توانیم با کلمات کلیدی (keyword) نیز نویسی کنیم.

```
print(2, 3) # positional
```

```
print(a=2, b=3) # keyword
```

```
print(b=3, a=2) # keyword
```

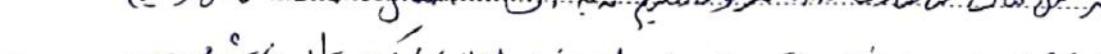
← ۱۶۱

def...parver(a=1,b=1): اسلیف بدلن رویهار که سیمین دستوره
 return a+b (return) (return) (return)
 print(parver(2,3)) #positional (return) (return) (return)
 print(parver(a=1,b=2)) #keyword (return) (return)
 کامیک هر دوی چونچ بری کامیک هر دوی کامیک keyword

* برای اظهاری مکرر **positional** از این شکل است که هر چیز قبل هدفی جست و با اینجاو که بینند.
 برای اظهاری مکرر **keyword** از این شکل است که هر چیز بعد هدفی جست و با اینجاو که بینند.
 درین ویژگی از **positional** و **keyword** اظهاری مکرر متفاوت است.

```
def func(a,b,c,d):
    print(a+b+d)
```

`def func(a, b, /, c):` دلیلیں کالج (اصلی) کے بعد میرے قبل فریڈ
`! positional اپارکی و مکاری بعد فریڈ`
`print(func(1, 2, d=5))` اپارکی کائن (مکاری و مکاری پہلے برابری کردی) keyword.

جلسہ ۱، ۲۹
تمہاری: UML = Unified Modeling Language (زبان مدل سازی کیا چکے) لئے (۱) اسے مذکور طبی UML (پر یقین دیکی، عاقلی کی مدل سازی کر) (۲) بچھی لزی مدل کریں (۳) UML سروکاری ترمیم کرنے آئے class diagram (جیوں) 

Bank Account	نکاح	نکاح
name	نام	نام
number	نمبر	نمبر
balance	بیلینس	بیلینس
deposit()	اڈٹریبیوٹ	اڈٹریبیوٹ
withdraw()	رکھار	رکھار
calculate-interest()	عملیات	عملیات

(ملاد استریو تله‌نگ هنوز درس را در خودش بروز دهد)

class BankAccount : class BankAccount ()
 pass
 #.bank مفهوم معمولی کلاس (فقط و باید) و صرف لعل و کلاس
 #.taxis با این کلاس ترکیب کنندگان بتوانند هر فرد (ولی
 #.tarif هر کدام خاصیت را که باشد (مثل: (Bank Account - Bank : (Random Number Format - Student - Random Number))
 #.tarif (Random Number Format - Student - Random Number))

BankAccount () مفهوم این کلاس که کلاس با مرور پذیر است یعنی بتوان
 Pass دفعه ۱. فقط این فرمات دسترسی دارد و پس از این نهاد.
 e1 (instance) که کلاس اس ای داری (Bank Account) می باشد BankAccount ☆
 print (BankAccount ()) آدرس مخفف این بروزروزی با جم خود را نماید.
 print (BankAccount ()) چون هر کدام بار که این دفعه ایجاد می شود این دفعه ایجاد می شود
 (مثلاً اولی متعلق به دریم و دی دوی متعلق به خیرالله)

b1 = BankAccount ()

b2 = BankAccount ()

print (type (b1), type (b2)) هردو (instance)

print (id (b1), id (b2)) آدرس های متن و سازمانی داشته باشند (بریم و خیرالله)

help (int) همچویست لطفاً استدعا کنید help : help ☆

{ print (isinstance (b1, BankAccount)) True : isinstance ☆

print (isinstance (b1, int)) False

(BankAccount ()) بجزی این که آن که داشته باشد باشند از شرایط برآورده باشند (بریم و خیرالله) (isinstance).

class BankAccount :

def deposit (self) : self = b1

return 'deposit'

b1 = BankAccount ()

print (b1.deposit ())

→ deposit

print (BankAccount.deposit (b1))

→ deposit

→ خطا

Subject: Congress (USA)
Year. 1981 Month. 1 Date. 29

```

class BankAccount:
    def deposit(self, name):
        print(name)
        return 'deposit'

bl = BankAccount()
print(bl.deposit('persian')) → {persian

```

تابع معادی که از آن برگزینی استفاده می‌کنیم ← function
 گنجی که درین کلاس هر چیزی را (عملیات، مقدارها، متد هایی...) نمایش دهد ← method
 (double under init) ←
 این کاملاً بعد از هر چیزی که باشد که درین کلاس نمایش داده شود ← init ←
 همان‌جا که در کل این کامب ← init ← استفاده می‌کنیم و هر چیزی را (مقدار، متد هایی...) نمایش دهد ← instance ←
 اگر این کامب ← (یعنی مثلاً هر کدام از BankAccount) ← init ← کامب ← instance ← BankAccount ← اینها را می‌توانیم
 b1 = BankAccount () ←? BankAccount ← init ← (b1)
 b1 = BankAccount () ←? BankAccount ← init ← (b1)
 چون باید (b1) ... instance ... init ... (b1) ... هستند
 self.balance = 0 ← (b1) ← مقدار (b1) که نمایش داده شود

```

class BankAccount :
    def __init__(self) :
        self.balance = 0
    def deposit(self, cash) :
        self.balance += cash
        return self.balance
bl = BankAccount()
print(bl.balance)
bl.deposit(50)
print(bl.balance)

```

#.data.append(l) (data.append(l)): ملکیت این مقوله را در میان دو کنونی داشته باشند.

#.inst.(data) (data.append): ملکیت این مقوله را در میان دو کنونی داشته باشند.

#.data + *

(استفاده از مجموعه ای از موارد این مقوله در میان دو کنونی)

class BankAccount:

این نکت مطابق با مفهوم خالق میگردد.

Def __init__(self):

self.balance = 0 این مکرر اولیه balance را

def deposit(self, cash):

balance را

self.balance += cash

balance را بروز رسانید.

return self.balance

def withdraw(self, cash):

self.balance -= cash

return self.balance

bl = BankAccount()

instantiation (ساخت) instance

print(bl.balance)

bl.deposit(100)

{ این مکرر اولیه میگیرد که مقدار 100 را در میان دو کنونی میگذارد.

bl.deposit(200)

{ این مکرر اولیه میگیرد که مقدار 200 را در میان دو کنونی میگذارد.

bl.withdraw(100)

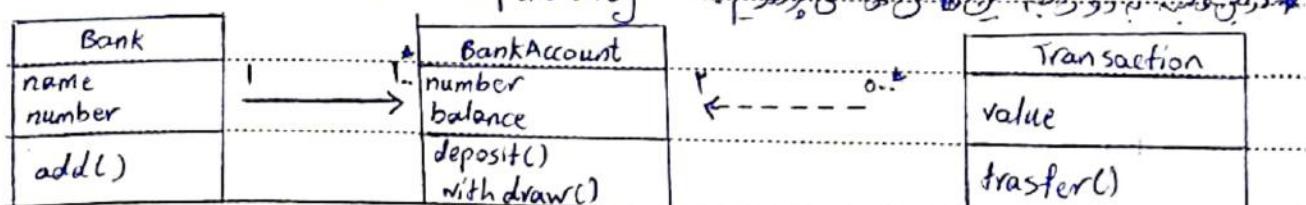
print(bl.balance) → 300

and bl = BankAccount() در

: پر کردن

این عبارت تولید یک جسم ممکن باشد (بالتالي ممکن است با این دستور مقدار 100 را در میان دو کنونی میگذارد).

association and dependency



UML نمودار

Dependency: برای عدم توانایی برخکسری بگزید و بسیار رود.
 همچنان بازهای این احتمال بوده و بعد از آن دلیر خاری باعث سُرمهگاری.
transfer(): معنی خواهد بود که از جهات مبادله این امثال در انتقال و چشم.

دستگیری (dependence) → میزان اعتمادی که یک فرد برای این اتفاق داشته باشد که در هر روزهای این اتفاق را بخواهد (association)

transfer(start, end)

transfer(p6, f6) transaction

class BankAccount:

def __init__(self, number, balance):

self.number = number

self.balance = balance

def deposit(self, value):

self.balance += value

def withdraw(self, value):

self.balance -= value

b1 = BankAccount(123, 0)

b2 = BankAccount(456, 100)

b1.transfer(b2, 50)

class Transaction:

def __init__(self, value):

self.value = value

def transfer(self, start, end):

start.withdraw(self.value)

end.deposit(self.value)

t1 = Transaction(10)

t1.transfer(b1, b2)

print(b1.balance) → 10

print(b2.balance) → 150

(balance = start.balance - value)

(balance = end.balance + value)

class Bank: (کلاس بانک) معرفی کردند که میتوانند این کلاس را با این نام در پایتون استفاده کنند.

```
def __init__(self, name, number):
    self.name = name
    self.number = number
    self.bank_accounts = []
    def add(self, bank_account):
        self.bank_accounts.append(bank_account)
    def remove(self, bank_account):
        self.bank_accounts.remove(bank_account)

parsian = Bank('Parsian', 1) یادداشت: این بخش از کد نشان میکند که یک اینستانس از کلاس بانک با نام 'parsian' و شماره '1' ایجاد شده است.
parsian.add(bl)
parsian.add(br)
print(parsian.bank_accounts) یادداشت: این بخش از کد نشان میکند که چهار اینستانس از کلاس بانک با نام 'bl', 'br', 'parsian' و 'melli' ایجاد شده اند و همگی در لیست 'parsian.bank_accounts' ذخیره شده اند.
```

parisian1 = Bank('Parsian', 1)
parisian2 = Bank('Parsian', 2)
melli1 = Bank('melli', 1) melli1.add(bl)
melli2 = Bank('melli', 2) melli2.add(br)
print(parsian1.bank_accounts) یادداشت: این بخش از کد نشان میکند که لیست 'parsian1.bank_accounts' شامل دو اینستانس از کلاس بانک با نام 'parsian1' و 'melli1' باشند.
print(parsian2.bank_accounts) یادداشت: این بخش از کد نشان میکند که لیست 'parsian2.bank_accounts' شامل دو اینستانس از کلاس بانک با نام 'parsian2' و 'melli2' باشند.
def get_bank(number):
 for bank in Banks:
 for bank_account in bank.bank_accounts:
 if number == bank_account.number:
 return bank
 print("Bank not found") یادداشت: این بخش از کد نشان میکند که اگر شماره مورد نظر در لیست 'Banks' وجود نداشته باشد، یک پیغام خطا نمایش داده شود.
b = get_bank(1)
print(b.name) یادداشت: این بخش از کد نشان میکند که نام اینستانس از کلاس بانک با شماره ۱ مطابقت داشته باشد.
print(b.number) یادداشت: این بخش از کد نشان میکند که شماره اینستانس از کلاس بانک با شماره ۱ مطابقت داشته باشد.

پس از این مراحل میتوان ساخت اولین رسم کن و کد نویسی

self.balance = balance لیکن با مرتبه بالاتر method پیش از self
 self.number = number مقدار سر و در مدخل عن آندر (فیل) بقایی نموده
 self.value = value که بعد از تعریف متغیرها باید مقدار داده شود. اگر کردی و بخواهید این را در کد خود ایجاد کنید این کار را باید انجام داد
که در کد زیر (است) که در فایل print باشد این کار انجام نشده است
 self.value = self.value که این اسکریپت self.value را بخواهد این کار را در اسکریپت print انجام داد
 self.value لیکن name error → The 'value' is not defined.

: ۲/۴

class یک رابطه فوکوس برگزالت
 class dependency association
 class association dependency dependency dependency dependency
 در واقع در هر چیزی که می توان آن را که استفاده می کند را بخواهید این را داشت
 * مثمر (object) که می توان آن را که استفاده می کند را بخواهید این را داشت
 (object) هر چیزی که می توان آن را که استفاده می کند را بخواهید این را داشت
 object که می توان آن را که استفاده می کند را بخواهید این را داشت

def که می توان آن را که استفاده می کند را بخواهید این را داشت
 (def که می توان آن را که استفاده می کند را بخواهید این را داشت
 dir(b1) که می توان آن را که استفاده می کند را بخواهید این را داشت
 bl که می توان آن را که استفاده می کند را بخواهید این را داشت
 bl که می توان آن را که استفاده می کند را بخواهید این را داشت
 type(bl) که می توان آن را که استفاده می کند را بخواهید این را داشت
 deposit که می توان آن را که استفاده می کند را بخواهید این را داشت
 withdraw که می توان آن را که استفاده می کند را بخواهید این را داشت
 (def که می توان آن را که استفاده می کند را بخواهید این را داشت
 bl که می توان آن را که استفاده می کند را بخواهید این را داشت
 bl که می توان آن را که استفاده می کند را بخواهید این را داشت
 bl که می توان آن را که استفاده می کند را بخواهید این را داشت

bl = BankAccount(1000, 20)

bl = BankAccount(1000, 20) که می توان آن را که استفاده می کند را بخواهید این را داشت
 def که می توان آن را که استفاده می کند را بخواهید این را داشت
 less than(bank-account), bank-account که می توان آن را که استفاده می کند را بخواهید این را داشت
 return که می توان آن را که استفاده می کند را بخواهید این را داشت
 print(less than(bl, bl)) → True
 print(less than(bl, b1)) → False

def lt_(self, other):
 if self < other:
 return True
 else:
 return self.balance > other.balance
print(bl < br) → True

def gt_(self, other):
 if self > other:
 return True
 else:
 return self.balance < other.balance
print(bl > br) → True

def eq_(self, other):
 return self.balance == other.balance
print(bl == br) → True

def neq_(self, other):
 return self.balance != other.balance
print(bl != br) → False

def ge_(self, other):
 if self >= other:
 return True
 else:
 return self.balance > other.balance
print(bl >= br) → True

f ① bl == 2 → True
② bl == br → False
③ bl == bl → True

def __repr__(self):
..... return f"BankAccount(number={self.number}, balance={self.balance})"
bl → BankAccount(number=123, balance=200)
در اینجا فرمت گذاری را تعریف کردند بنابراین
در اینجا در لستینگ های BankAccount که خواهد داشت این را داشتند:
{self.__class__.__name__}

برای این ساختار self باید در under class باشد
که اینجا با عنوان bl نوشته شده است.
با این کشور در هر کشوری (class.BankAccount) و دست را تعمیر دهیم که در کشور ایالات متحده آمریکا نیز نوشته شده است.
این کشوری کشور ایالات متحده آمریکا فروغی نداشتن از دعوهای رایجنهای ایالات متحده است.
در اینجا از under str برای under repr استفاده نموده اند. این اتفاق ممکن است در هر کشوری که از under str استفاده نموده باشد اینجا نیز از under str استفاده نموده باشد.
پس از اینجا under str, under repr, class.__name__, under str تو
* در اینجا در هر کشوری که ایالات متحده نباشد:

- ① inst(data) → function(object)
- ② data.inst(parameter) → object.method(parameter)
- ③ data* → object.method_(parameter)

هر ۳ روش در کتابخانه (جهانی - تحریری) در هر کشوری متفاوت است.
(مورد اولم طوری می‌گذرد که باید اسازی کارکردی ریکالت انجام نماید. حالا از
def __getitem__(self, item):
..... return self.bank_accounts[item] → قدراینها امتحان به است. قدراینها امتحان به است.

def __len__(self):
..... return len(self.bank_accounts) → under len if class

* دلار - dollar - getitem - ابتدا دینم - می براحت تابع خود را به تابع لست تبدیل کرد و در ای کن همان حالت پردازند.

```
{for item in bank!:
    print(item)   # فرمی
    balance = item.balance
    number = item.number
    print(f'Name: {item.name} - Balance: {item.balance} - Account Number: {item.number}')}
```

* مثل) این فرآیند موجودی فعل دو بانک را با جم کرده است - موجودی فعل در اینجا با معنای موجودی فعل ای دیگر بانک است که این بانک این موجودی داشته باشد:

```
def __str__(self, other):
    return self.total_balance() + other.total_balance()

def total_balance(self):
    self.balance = 0
    for bank_account in self.bank_accounts:
        self.balance += bank_account.balance
    return self.balance
```

حالت دیگر

* دلاری پاسخ - under - دلاری پاسخ

* دلاری مقامیه (دو فرآیند پایه داری دارد) - get - it -

* دلاری (که دلاری دارد) - str -

* دلاری (که دلاری دارد) - developer - str -

bl = Bank('Persian', rta)

(bl

bal = BankAccount(111111, 100)

bar = BankAccount(111111, 100)

bl.append(bal)

bl.append(bar)

print(bl, bl.total_balance())

→ [BankAccount(number=111111, balance=100), BankAccount(number=111111, balance=100)]

for bank_account in bl:

print(bank_account) → دلاری داشتن ای دلاری داشتن

(دستور دلاری داشتن)

← ۱۷۱

Subject: Urdu - English
Year 1961 Month IV Date W ()

این بخش تمرینی در مورد دستور `return` است که در متaprogramming مفید است. مثلاً می‌توان از آن برای محاسبه مساحت یک مربع استفاده کرد.

```
def area(l, w):
    return l * w
```

برای این کار باید ابتدا مساحت را با عنوان `area` نویسیم و آنرا در کلاس `rectangle` ذکر کنیم.

```
class rectangle:
    def area(self):
        return self.length * self.width
```

با این کار می‌توان از کلاس `rectangle` یک اینسنسیتی (instance) ایجاد کرد و مساحت آن را محاسبه کرد.

```
r = rectangle()
print(r.area())
```

در اینجا `r` یک اینسنسیتی (instance) است که متعلق به کلاس `rectangle` است. این کلاس مساحت را با عنوان `area` نویسیم و آنرا در کلاس `rectangle` ذکر کنیم.

```
class rectangle:
    def area(self):
        return self.length * self.width
```

با این کار می‌توان از کلاس `rectangle` یک اینسنسیتی (instance) ایجاد کرد و مساحت آن را محاسبه کرد.

```
r = rectangle()
print(r.area())
```