

Subject:

پایتون یک زبان تفسیری مدرن، مناسب
جای سافت برنامه‌های با سایز متوسط.

برنامه‌های پایتون را کجا می‌نویسیم؟
فایل متنی یا پرسون Py.

Source Code را کجا اجرا کنیم.

به عنوان ورودی به مفسر Python می‌دهیم.

Python code.py

معرفی Syntax زبان پایتون.

متغیرها: اعلان ندارند، نیاز به
تعیین نوع ندارند.

a = 123
name = 'string'

IDEA

2

Subject:

ساختار شرطی.

if logical-expression:
statement 1
statement 2

⋮
statement n

elif logical-expression:

else:

⋮

اختیاری

ساختار تکرار

while logical-expression:
statement 1

⋮
statement n.

for variable in collection:
loops for each item in collection.
⋮

3

Subject:

logical expressions:

$a \wedge b \rightarrow \text{Python: } a \text{ and } b$

$a \vee b \rightarrow \text{Python: } a \text{ or } b$

$\neg a \rightarrow \text{Python: } \text{not } a$

functions.

توابع

`def (Parameter 1, Parameter 2, ...)`

`Statement 1`

`Statement 2`

`:`

`[return value` اختیاری

ماژول ها

چگونه از ماژول ها استفاده کنیم

`import module-name`

4

Subject:

چگونه از اجزای ماژول استفاده کنیم

`module-name . defined-name`

در هر ماژول چه اجزایی قرار دارد

`dir (module-name)`

هر یک از توابع و مقادیر از پیش تعیین شده چگونه کار می کنند

`help (defined-name)`

کجا درباره ی Python مطالعه کنیم

www.docs.python.org

پایتون 2 و پایتون 3

تفاوت بین Python 2.x و Python 3.x

های قابل توجهی وجود دارد

5

Subject:

ساختار داده‌ها در Python

روش‌های دسته‌بندی و پردازش اطلاعات
و نمایش نحوه‌ی ارتباط داده‌ها.

چرا از ساختار داده‌ها استفاده کنیم؟
• خوانایی کد

• تسهیل نمایش ارتباط بین داده‌ها
• قابلیت استفاده از مدل‌های
قدرتمند از پیش ساخته شده.

چه ساختار داده‌هایی؟

String •

list •

tuple •

Dictionary •

Set •

امروز:

Strings (رشته‌ها) •

lists (لیست‌ها) •