Lecture 27: Regular Expressions and Python Identifiers
Python syntax makes very few restrictions on the ways that we can name our variables, functions, and classes.

- Variables names must start with a letter or an underscore
- Every character after the first (if any) must be a letter, underscore, or number
- Names cannot be a reserved Python keyword:

  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
Python syntax makes very few restrictions on the ways that we can name our variables, functions, and classes.

- Variables names must start with a letter or an underscore
- Every character after the first (if any) must be a letter, underscore, or number
- Names cannot be a reserved Python keyword:
  
  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  

But Python’s PEP 8 (Python Enhancement Proposal 8) specifies, in detail, a list of naming conventions.

- These are not enforced by the language. They are enforced by us as programmers.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter)

- **B** (single uppercase letter)

- **lowercase**

- **lower_case_with_underscores**

- **UPPERCASE**

- **UPPER_CASE_WITH_UNDERSCORES**

- **CapitalizedWords** (or CapWords, or CamelCase, or StudlyCaps).

- **mixedCase** (differs from CapitalizedWords by initial lowercase character!)

- **Capitalized_Words_With_Underscores**

We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter)  
  \[a-z]\n
- **B** (single uppercase letter)

- **lowercase**

- **lower_case_with_underscores**

- **UPPERCASE**

- **UPPER_CASE_WITH_UNDERSCORES**

- **CapitalizedWords** (or **CapWords**, or **CamelCase**, or **StudlyCaps**).

- **mixedCase** (differs from **CapitalizedWords** by initial lowercase character!)

- **Capitalized_Words_With_Underscores**

We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter) 
  
  `[a-z]`

- **B** (single uppercase letter) 
  
  `[A-Z]`

- **lower**
  
  `lower_case_withunderscores`

- **UPPERCASE**
  
  `UPPER_CASE_WITH_UNDERSCORES`

- **CapitalizedWords** (or **CapWords**, or **CamelCase**, or **StudlyCaps**).
  
  `CapitalizedWords`

- **mixedCase** (differs from **CapitalizedWords** by initial lowercase character!)
  
  `mixedCase`

- **Capitalized_Words_With_Underscores**

We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter) 
  \[a-z]\n
- **B** (single uppercase letter) 
  \[A-Z]\n
- **lowercase** 
  \[a-z]+\n
- **lower_case_with_underscores**

- **UPPERCASE**

- **UPPER_CASE_WITH_UNDERSCORES**

- **CapitalizedWords** (or CapWords, or CamelCase, or StudlyCaps).

- **mixedCase** (differs from CapitalizedWords by initial lowercase character!)

- **Capitalized_Words_With_Underscores**

We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter)  
  
- **B** (single uppercase letter)  
  
- **lowercase**  
  
- **UPPERCASE**  
  
- **UPPER_CASE_WITH_UNDERSCORES**  
  
- **CapitalizedWords** (or CapWords, or CamelCase, or StudlyCaps).  
  
- **mixedCase** (differs from CapitalizedWords by initial lowercase character!)  
  
- **Capitalized_Words_With_Underscores**  
  
We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter) 
  \[a-z]\n
- **B** (single uppercase letter) 
  \[A-Z]\n
- **lowercase** 
  \[a-z]+\n
- **lower_case_with_underscores** 
  \[a-z][a-z.]*\n
- **UPPERCASE** 
  \[A-Z]+\n
- **UPPER_CASE_WITH_UNDERSCORES**

- **CapitalizedWords** (or **CapWords**, or **CamelCase**, or **StudlyCaps**).

- **mixedCase** (differs from **CapitalizedWords** by initial lowercase character!)

- **Capitalized_Words_With_Underscores**

We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter) 
  \[a-z]\]
- **B** (single uppercase letter) 
  \[A-Z]\]
- **lowercase** 
  \[a-z]+\]
- **lower_case_with_underscores** 
  \[a-z][a-z_.]*\]
- **UPPERCASE** 
  \[A-Z]+\]
- **UPPER_CASE_WITH_UNDERSCORES** 
  \[A-Z][A-Z_]*\]
- **CapitalizedWords** (or CapWords, or CamelCase, or StudlyCaps).
- **mixedCase** (differs from CapitalizedWords by initial lowercase character!)
- **Capitalized_Words_With_Underscores**

We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter)
  
  [a-z]

- **B** (single uppercase letter)
  
  [A-Z]

- **lowercase**
  
  [a-z]+

- **lower_case_with_underscores**
  
  [a-z][a-z-]*

- **UPPERCASE**
  
  [A-Z]+

- **UPPER_CASE_WITH_UNDERSCORES**
  
  [A-Z][A-Z-]*

- **CapitalizedWords** (or CapWords, or CamelCase, or StudlyCaps).
  
  [A-Z][a-zA-Z]*

- **mixedCase** (differs from CapitalizedWords by initial lowercase character!)
  
  [a-z][a-zA-Z]*

- **Capitalized_Words_With_Underscores**

We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter)  
  
- **B** (single uppercase letter)  
  
- **lowercase**  
  
- **UPPERCASE**  
  
- **lower_case_with_underscores**  
  
- **UPPER_CASE_WITH_UNDERSCORES**  
  
- **CapitalizedWords** (or **CapWords**, or **CamelCase**, or **StudlyCaps**).  
  
- **mixedCase** (differs from **CapitalizedWords** by initial lowercase character!)  
  
- **Capitalized_Words_With_Underscores**  

We can define regular expressions for each.
Other languages also have naming conventions. Popular styles include:

- **b** (single lowercase letter)
  - `[a-z]`

- **B** (single uppercase letter)
  - `[A-Z]`

- **lowercase**
  - `[a-z]+`

- **lower_case_with_underscores**
  - `[a-z][a-z_]*`

- **UPPERCASE**
  - `[A-Z]+`

- **UPPER_CASE_WITH_UNDERSCORES**
  - `[A-Z][A-Z_]*`

- **CapitalizedWords** (or CapWords, or CamelCase, or StudlyCaps).
  - `[A-Z][a-zA-Z]*`

- **mixedCase** (differs from CapitalizedWords by initial lowercase character!)
  - `[a-z][a-zA-Z]*`

- **Capitalized_Words_With_Underscores**
  - `[A-Z][a-zA-Z_]*`

We can define regular expressions for each.
Python specifies specific naming conventions depending on the usage:

- **Class names**:
- **Exceptions**: local variables:
- **Function names**:
- **Instance variables (public)**:
- **Instance variables (private)**:
- **Constants**:
- **Modules**:
- “Magic” object or attribute:
Python specifies specific naming conventions depending on the usage:

- **Class names**: `[A-Z][0-9a-zA-Z]*`
- **Exceptions**: `[A-Z][0-9a-zA-Z]*Error`
- **Local variables**: `[a-z][0-9a-z]*`
- **Function names**: `[a-z][0-9a-z]*`
- **Instance variables (public)**: `self.[a-z][a-z0-9]*`
- **Instance variables (private)**: `self.[a-z][a-z0-9]*`
- **Constants**: `[A-Z][A-Z0-9]*`
- **Modules**: `[a-z][0-9a-z]{0,15}`
- **“Magic” object or attribute**: `([a-z]|([a-z][a-z0-9]*[a-z0-9]))`
Python specifies specific naming conventions depending on the usage:

- **Class names**: [A-Z][0-9a-zA-Z]*
- **Exceptions**: [A-Z][0-9a-zA-Z]*Error
- **local variables**: __
- **Function names**: __
- **Instance variables (public)**: __
- **Instance variables (private)**: __
  - **Constants**: __
  - **Modules**: __
- **“Magic” object or attribute**: __
Python specifies specific naming conventions depending on the usage:

- **Class names**: [A-Z][0-9a-zA-Z]*
- **Exceptions**: [A-Z][0-9a-zA-Z]*Error
- **Local variables**: [a-z][0-9a-z]*
- **Function names**: 
- **Instance variables (public)**: 
- **Instance variables (private)**: 
  - **Constants**: 
  - **Modules**: 
- **“Magic” object or attribute**: 
Python specifies specific naming conventions depending on the usage:

- **Class names**: [A-Z][0-9a-zA-Z]*
- **Exceptions**: [A-Z][0-9a-zA-Z]*Error
- **Local variables**: [a-z][0-9a-z_]*
- **Function names**: [a-z][0-9a-z_]*
- **Instance variables (public)**:
- **Instance variables (private)**:
  - **Constants**:
  - **Modules**:
- **“Magic” object or attribute**: 
Python specifies specific naming conventions depending on the usage:

- **Class names**: `[A-Z][0-9a-zA-Z]*`
- **Exceptions**: `[A-Z][0-9a-zA-Z]*Error`
- **Local variables**: `[a-z][0-9a-z]*`
- **Function names**: `[a-z][0-9a-z]*\(`
- **Instance variables (public)**: `self\.[a-z][a-z0-9]*`
- **Instance variables (private)**:
  - **Constants**:
  - **Modules**:
- **“Magic” object or attribute**:
Python specifies specific naming conventions depending on the usage:

- **Class names**: `[A-Z][0-9a-zA-Z]*`
- **Exceptions**: `[A-Z][0-9a-zA-Z]*Error`
- **Local variables**: `[a-z][0-9a-z_]`
- **Function names**: `[a-z][0-9a-z_]`
- **Instance variables (public)**: `self\.[a-z][a-z0-9_]`
- **Instance variables (private)**: `self\_.[a-z][a-z0-9_]`
- **Constants**:
- **Modules**:
- **“Magic” object or attribute**:
Python specifies specific naming conventions depending on the usage:

- **Class names**: [A-Z][0-9a-zA-Z]*
- **Exceptions**: [A-Z][0-9a-zA-Z]*Error
- **local variables**: [a-z][0-9a-z_]*
- **Function names**: [a-z][0-9a-z_]*\(\)
- **Instance variables (public)**: self\.[a-z][a-z0-9_]*
- **Instance variables (private)**: self\._[a-z][a-z0-9_]*
  
- **Constants**: [A-Z][A-Z0-9_]*
- **Modules**:
- **“Magic” object or attribute**:
Python specifies specific naming conventions depending on the usage:

- **Class names**: `[A-Z][0-9a-zA-Z]*`
- **Exceptions**: `[A-Z][0-9a-zA-Z]*Error`
- **Local variables**: `[a-z][0-9a-z_]`
- **Function names**: `[a-z][0-9a-z_]\(`
- **Instance variables (public)**: `self\.[a-z][a-z0-9_.]*`
- **Instance variables (private)**: `self\.[a-z][a-z0-9_.]*`
- **Constants**: `[A-Z][A-Z0-9_.]*`
- **Modules**: `[a-z][0-9a-z_]{0,15}`
- **“Magic” object or attribute**: 
Python specifies specific naming conventions depending on the usage:

- **Class names**: `[A-Z][0-9a-zA-Z]*`
- **Exceptions**: `[A-Z][0-9a-zA-Z]*Error`
- **Local variables**: `[a-z][0-9a-z_]`
- **Function names**: `[a-z][0-9a-z_]`
- **Instance variables (public)**: `self\.\[a-z][a-z0-9_]`
- **Instance variables (private)**: `self\.\.[a-z][a-z0-9_]`
- **Constants**: `[A-Z][A-Z0-9_]`
- **Modules**: `[a-z][0-9a-z_]`, `{0,15}
- **“Magic” object or attribute**: `(__[a-z]__)|(__[a-z][a-z0-9_]*)[a-z0-9]__`