

PEP 308: The Implementation of if, else in Python

CSCI 3155

December 17, 2013

Brief History of Python

PEP 308: The
Implementa-
tion of `if`,
`else` in
Python

CSCI 3155

- Implemented in 1989 by Guido van Rossum
- liked by many for its concise syntax, ease of use and simple to learn
- Believe it or not! Python did not have a conditional expression feature until 2005

Conditional Expressions

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- allows a language to perform an action based on some condition
- the input is evaluated to true or false based on the condition
- `if`, `else` statement `if(Neo) "the one" else "agent"`

Previous to Python 2.5

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- before Python 2.5 was released, users resorted to `and`, `or` statements
- `and` or
- worked, but often caused for errors

In Comparison

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

```
X = <condition>  
Y = <expression1>  
Z = <expression2>
```

```
X and Y or Z  
Y if X else Z
```

```
neo and "the one" or "agent"  
"the one" if (neo) else "agent"
```

PEP 308

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- `and`, `or` resulted in many errors
- thus leading to Python Enhancement Proposal 308

Where was if-else?

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- Lack of consensus for syntax delayed implementation
- Language still young and being developed
- The current way worked well enough

Original Proposal

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- Original version was

`<expression1> if <condition> else <expression2>`

- Confusing out of order arrangement
- Conditional could easily be skipped over

Discussion and... No agreement

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- Discussions about the implementation occurred on both *python-dev* and *comp.lang.python*
- No one could agree, developer Guido van Rossum simply chose this:

```
x = true_value if condition else false_value
```

Issues with Chosen Syntax

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

```
x = true_value if condition else false_value
```

- Syntax can be somewhat confusing
- Could either see it as `x = true_value else false_value`
- Evaluation still lazy, as Boolean expressions are

Application to Principles of Programming Languages

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- many applications!
 1. syntax
 2. semantics
 3. parsing
 4. Maybe even tail recursion?

Syntax

PEP 308: The
Implementation of if,
else in
Python

CSCI 3155

- structure of the expressions in the language
- Current example of if/else python syntax

```
if x < 0:
    x = 0
    print 'Negative changed to zero'
elif x == 0:
    print 'Zero'
elif x == 1:
    print 'Single'
else:
    print 'More'
```

- standardized what the if, else expression was supposed to look like
- did not include parenthesis around conditional because it wasn't python style

Semantics

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- meaning of the syntax of the language
- Documented what `if`, `else` was supposed to mean
- standardized when the `if`, `else` expression was supposed to be used

Parsing

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- required new evaluation judgement forms
- changes to the abstract syntax tree cases
- required new type checking and inference
- all this to keep the syntax unambiguous

Tail Recursion

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- `if`, `else` acts like case matching in `scala`
- could be used in place of `switch-case` to create tail recursion within methods
- **but** python does not support tail recursion
- Guido thought it would detract from python's dynamic nature
- impossible to do certain compile-time optimizations

Detractors of PEP 308

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

Group 1: *Wanted to adopt a ternary operator built using punctuation characters*

`<condition> ? <expression1> : <expression2>`

- more concise
- simplistic and intuitive
- prone to confusion of definitions

More Detractors

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

Group 2: *Wished to adopt a ternary operator built using new or existing keywords*

```
<condition> then <expression1> else <expression2>  
(if <condition> : <expression1> else: <expression2>)
```

- didn't wish to include new syntax in Python
- aimed to avoid affecting backwards compatibility
- further confusion and double defining

More Detractors

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

Group 3:

DO NOTHING.

- used previous methods of `and` and `or` to implement logic
- avoided all possible errors including backwards compatibility
- backlogged performance improvement and updates

Acceptance of PEP 308

- Guido van Rossum accepts proposal
 - Friday, September 30th, 2005
 - a sarcastic yet fitting letter to naysayers
- Implemented soon after into the now familiar `if`, `else`

Conclusion

PEP 308: The
Implementation of `if`,
`else` in
Python

CSCI 3155

- Once `if`, `else` implemented, talk of negative impact dissipated
- Improved ease of conditionals
- Python finally started to catch up to the base level of competitors
- Winning.