What are your language "hangups"? [closed]

Asked 16 years, 2 months ago Modified 12 years, 10 months ago

Viewed 2k times

16

votes

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, visit the help center for guidance.

Closed 13 years ago.

Locked. This question and its answers are locked because the question is off-topic but has historical significance. It is not currently accepting new answers or interactions.

I've read some of the recent language vs. language questions with interest... Perl vs. Python, Python vs. Java, Can one language be better than another?

One thing I've noticed is that a lot of us have *very superficial* reasons for disliking languages. We notice these things at first glance and they turn us off. We shun what are probably perfectly good languages as a result of features that we'd probably learn to love or ignore in 2 seconds if we bothered.

Well, I'm as guilty as the next guy, if not more. Here goes:

- Ruby: All the Ruby example code I see uses the `puts` command, and that's a sort of childish Yiddish anatomical term. So as a result, I can't take Ruby code seriously even though I should.
- Python: The first time I saw it, I smirked at the whole significant whitespace thing. I avoided it for the next several years. Now I hardly use anything else.
- Java: I don't like identifiersThatLookLikeThis. I'm not sure why exactly.
- Lisp: I have trouble with all the parentheses. Things of different importance and purpose (function declarations, variable assignments, etc.) are not syntactically differentiated and I'm too lazy to learn what's what.
- Fortran: uppercase everything hurts my eyes. I know modern code doesn't have to be written like that, but most example code is...
- Visual Basic: it bugs me that `Dim` is used to declare variables, since I remember the good ol' days of GW-BASIC when it was *only* used to dimension arrays.

What languages *did* look right to me at first glance? Perl, C, QBasic, JavaScript, assembly language, BASH shell, FORTH.

Okay, now that I've aired my dirty laundry... I want to hear yours. **What are your language hangups? What superficial features bother you? How have you gotten over them?**

syntax language-features language-design

Share

edited May 23, 2017 at 12:14

community wiki
2 revs
Dan Lenski

I agree about Python - except I'm still avoiding it. I'd have avoided Haskell too - except it lets you override the whitespace and use curly brackets, which I always do. – finnw Oct 8, 2008 at 22:26

Well I gotta reiterate that I'm extremely glad I got over my Python whitespace hangup. :) – Dan Lenski Oct 9, 2008 at 0:41

Well, Python whitespace use does seem to impose limitations: why can lambda take only one statement? – Svante Nov 21, 2008 at 16:45

2 Isn't the Yiddish term 'putz'? I've never seen it written as 'puts'. I guess I'm saying that I would never have thought to pronounce

'puts' that way. – Telemachus Feb 22, 2009 at 20:33

1 Puts is certainly pronounced differently than putz. The 'u' sounds significantly different in both words. – Dean J Dec 2, 2009 at 17:48

Comments disabled on deleted / locked posts / reviews |

37 Answers

Sorted by: Highest score (default)

1 2 Next

32 votes

I hate Hate HATE "End Function" and "End IF" and "If... Then" parts of VB. I would much rather see a curly bracket instead.

Share

answered Oct 2, 2008 at 5:56

community wiki
Wayne

At least it has one benefit - you can see what construct is ending without having to search for a matching bracket which might be off the top of the screen. – Hugh Allen Oct 25, 2008 at 13:12

Multiple nested END IFs in VB would have the same problem... I tend to use comments on the closing brace in code with many nested blocks, like "// end for y", etc. – Jared Updike Oct 29, 2008 at 18:57

I hate END Funtions too. I prefer bash syntax (if/fi, case/esac) or curly braces... – Seiti Nov 3, 2008 at 1:55

Yeah, I really hate how verbose VB is. – cdmckay Feb 22, 2009 at 20:35

14 Seiti: You would want to see NOITCNUF at the end of a function? And ERUDECORP? and SSALC? How about ECAFRETNI? For POO (er, I mean OOP) languages? – jmucchiello Sep 18, 2009 at 19:39

22 votes

PHP's function name inconsistencies.

```
// common parameters back-to-front
in_array(needle, haystack);
strpos(haystack, needle);

// _ to separate words, or not?
filesize();
file_exists;

// super globals prefix?
$GLOBALS;
$_POST;
```

Share

answered Oct 2, 2008 at 7:20

community wiki
Dean Rather

Wow, I just read my example comment "// _ to separate words, or not?" and realized the word "separate" - although always having the same meaning - is pronounced differently in 2 different contexts. One is like an assignment and the other is a comparator... I suppose English is a pretty bad too. – Dean Rather Oct 2, 2008 at 7:54

Yeah, I don't understand why they don't they just keep all the functions that pollute the global namespace (i.e. strpos, etc.) but also include new classes that clean them up (like a String class, a Number class, etc.) – cdmckay Feb 22, 2009 at 20:37

For that matter, Python's naming inconsistencies. That's one thing I will give Java. EVERYTHING in Java is named "right." – Instance Hunter Feb 28, 2009 at 5:40

4 Except that sometimes it's `.length`, and sometimes it's `.length()`, and yet other times it's `.getLength()` ~ – Pavel Minaev Dec 1, 2009 at 23:19

If only naming was the worst of PHP's inconsistencies! – EMP Dec 2, 2009 at 0:27

19

votes

I never really liked the keywords spelled backwards in some scripting shells

if-then-**fi** is bad enough, but case-in-**esac** is just getting silly

Share

answered Oct 2, 2008 at 7:57

community wiki
loudej

2 Heh, I have to agree with you on this one... and why do for and while loops end with "done" rather than "rof" or "elihw" :-P
– Dan Lenski Oct 2, 2008 at 15:59

I so want a language where `rofl` is a keyword – Earlz Dec 1, 2009 at 23:49

12

votes

I just thought of another... I hate the mostly-meaningless URLs used in XML to define namespaces, e.g. `xmlns="http://purl.org/rss/1.0/"`

Share

answered Oct 2, 2008 at 16:56

community wiki
Dan Lenski

I actually like this decision. URL's are unlikely to collide (Since you must actually purchase the domain name) and it provides a meaningful pointer for where you should go to look about the meaning of the tags in that namespace (though there's no requirement that the resource actually exists.) – SingleNegationElimination Apr 22, 2011 at 12:58

9 votes

Pascal's `Begin` and `End`. Too verbose, not subject to bracket matching, and worse, there isn't a `Begin` for every `End`, eg.

```
Type foo = Record
    // ...
end;
```

Share

answered Oct 2, 2008 at 6:23

community wiki
Hugh Allen

1 Any Pascal/Delphi IDE worth its salt will bracket-match `begin..end`, and a lot of advanced plain text editors do it as well. Mine does (it also bracket-matches `If...Then..End If` in BASIC, etc). – Pavel Minaev Dec 1, 2009 at 23:19

9 votes

Although I'm mainly a PHP developer, I dislike languages that don't let me do enough things inline. E.g.:

```
$x = returnsArray();
$x[1];
```

instead of

```
returnsArray()[1];
```

or

```
function sort($a, $b) {
    return $a < $b;
}
usort($array, 'sort');
```

instead of

```
usort($array, function($a, $b) { return $a < $b;
});
```

Share

answered Oct 2, 2008 at 6:27

community wiki
deceze

2 The latter is doable in php 5.3 – moo Dec 1, 2009 at 23:59

and how about `$(returnsArray())[1]` ?
– Radagast the Brown Mar 26, 2011 at 14:26

8
votes

I like object-oriented style. So it bugs me in Python to see len(str) to get the length of a string, or splitting strings like split(str, "|") in another language. That is fine in C; it doesn't have objects. But Python, D, etc. do have objects and use obj.method() other places. (I still think Python is a great language.)

Inconsistency is another big one for me. I do not like inconsistent naming in the same library: length(), size(), getLength(), getlength(), toUTFindex() (why not toUtfIndex?), Constant, CONSTANT, etc.

The long names in .NET bother me sometimes. Can't they shorten DataGridViewCellContextMenuStripNeededEventArgs somehow? What about ListViewVirtualItemsSelectionRangeChangedEventArgs?

And I hate deep directory trees. If a library/project has a 5 level deep directory tree, I'm going to have trouble with it.

Share

answered Feb 22, 2009 at 21:37

community wiki
Jordan Miner

4 Seriously. Why make it 99% object-oriented and then throw some random non-OO stuff in there for no apparent reason? GAH! – Instance Hunter Feb 28, 2009 at 5:44

I so identify with you. For some obscure reason I find myself now in java - where web projects **starts** with 5 levels dir-tree - before you get to write a single line of code... – Radagast the Brown Mar 11, 2011 at 21:49

one of the most irritating things in any OO language.
– Roja Buck Apr 14, 2011 at 11:47

7

votes

C and C++'s syntax is a bit quirky. They reuse operators for different things. You're probably so used to it that you don't think about it (nor do I), but consider how many meanings parentheses have:

```
int main()          // function declaration /
definition
printf("hello")     // function call
(int)x              // type cast
2*(7+8)             // override precedence
int (*)(int)        // function pointer
int x(3)            // initializer
if (condition)      // special part of syntax of if,
while, for, switch
```

And if in C++ you saw

```
foo<bar>(baz(),baaz)
```

you couldn't know the meaning without the definition of `foo` and `bar`.

- the < and > might be a template instantiation, or might be less-than and greater-than (unusual but legal)
- the () might be a function call, or might be just surrounding the comma operator (ie. perform baz() for size-effects, then return baaz).

The silly thing is that other languages have copied some of these characteristics!

Share

edited Oct 2, 2008 at 7:58

community wiki
2 revs
Hugh Allen

1 Consistent and smart use of whitespace can help clear up some of the ambiguities... but no one cares. – yodie Dec 2, 2009 at 0:05

1 @yodie: except, that it doesn't. The whitespace might suggest what the author meant, but it doesn't inform the compiler to actually *do* what the author tried to type. – SingleNegationElimination May 17, 2011 at 2:13

Right. It clears it up for the readers (including the author), but not for the compiler. Still, I find that my message tends to be clearer to the compiler when I format correctly, because I'll catch my mistakes (or won't make them). – yodie May 17, 2011 at 14:11

6 votes

Java, and its checked exceptions. I left Java for a while, dwelling in the .NET world, then recently came back.

It feels like, sometimes, my `throws` clause is more voluminous than my method content.

Share

answered Oct 2, 2008 at 5:25

community wiki
Michael Petrotta

3 The java exception constructor has a "cause" parameter so that you can do exception chaining, precisely to prevent this. You should only throw exceptions which are relevant at the abstraction level of your class. – Wim Coenen Feb 22, 2009 at 20:34

6

votes

There's nothing in the world I hate more than php.

1. Variables with $, that's one extra odd character for every variable.
2. Members are accessed with -> for no apparent reason, one extra character for every member access.
3. A freakshow of language really.
4. No namespaces.
5. Strings are concatenated with ..
6. A freakshow of language.

Share

answered Oct 2, 2008 at 7:38

community wiki
Vasil

Now that I think it, . can't be used instead of -> because it's concatenation... mmm... – asterite Oct 2, 2008 at 17:08

1 And the extra characters ($ and >) require you press shift which is annoying and increases typos. – Jared Updike Oct 29, 2008 at 18:59

1 not much, if you don't think operators in a programming language should be intuitive. Most proper OO languages have . for member access and + for string concatenation. So you'd expect similar behaviour. But then, php is not an OO language, let alone a proper language. – Vasil Feb 23, 2009 at 21:09

1 What is a "proper" OO language? There are plenty of OO language that use neither . nor -> to access data members. And what is a "proper" language? Show me a definition. Brainfuck is extremely consistent. Is that a "proper" language? – jmucchiello Sep 18, 2009 at 19:50

I would suggest that in languages which have both mutable value types and mutable reference types, having different symbols for member access is probably a good thing. – supercat Aug 29, 2011 at 2:44

5 votes

All the []s and @s in Objective C. Their use is so different from the underlying C's native syntax that the first time I saw them it gave the impression that all the object-orientation had been clumsily bolted on as an afterthought.

Share

answered Feb 28, 2009 at 5:31

community wiki
Crashworks

It freaks me out, it's C, but it aint grandpa's C – seanb Dec 2, 2009 at 1:02

3 It was bolted on as an afterthought. It started basically as a preprocessor for C. The @'s were there because @ was not legal for anything else, so they could be sure that any new keyword starting with @ (@interface, @implementation, etc.) would not conflict with any C keywords. Objective-C is terribly ugly, which is unfortunate, because the underlying object-oriented model and frameworks are pretty nice.
– Kristopher Johnson Jun 28, 2010 at 17:04

5

votes

I abhor the **boiler plate verbosity of Java**.

- writing getters and setters for properties
- checked exception handling and all the verbiage that implies
- long lists of imports

Those, in connection with the Java convention of using `veryLongVariableNames`, sometimes have me thinking I'm back in the 80's, writing `IDENTIFICATION DIVISION.` at the top of my programs.

Hint: If you can automate the generation of part of your code in your IDE, that's a good hint that you're producing boilerplate code. With automated tools, it's not a problem to write, but it's a hindrance every time someone has to read that code - which is more often.

While I think it goes a bit overboard on type bureaucracy, Scala has successfully addressed some of these concerns.

Share

answered Dec 1, 2009 at 23:43

community wiki
Carl Smotricz

3 `System.out.Println` is ridiculous for what most languages call `puts` – Earlz Dec 1, 2009 at 23:58

You forgot the depth of directory trees - and the str1.equals(str2), oh, it doesn't have an end :{ – Radagast the Brown Mar 11, 2011 at 21:51

3 @Earlz at least it's consistent with OO calling methodology. In java's case puts would have to be either a keyword or a static method on Object. The cost of carrying that method may well be high. – Roja Buck Apr 14, 2011 at 11:51

4 votes

Coding Style inconsistencies in team projects.

I'm working on a large team project where some contributors have used 4 spaces instead of the tab character. Working with their code can be very annoying - I like to keep my code clean and with a consistent style.

It's bad enough when you use different standards for different languages, but in a web project with HTML, CSS, Javascript, PHP and MySQL, that's 5 languages, 5 different styles, and multiplied by the number of people working on the project.

I'd love to re-format my co-workers code when I need to fix something, but then the repository would think I changed every line of their code.

Share

answered Oct 2, 2008 at 13:55

community wiki
Dean Rather

3

votes

It irritates me sometimes how people expect there to be one language for all jobs. Depending on the task you are doing, each language has its advantages and disadvantages. I like the C-based syntax languages because it's what I'm most used to and I like the flexibility they tend to bestow on the developer. Of course, with great power comes great responsibility, and having the power to write 150 line LINQ statements doesn't mean you should.

I love the inline XML in the latest version of VB.NET although I don't like working with VB mainly because I find the IDE less helpful than the IDE for C#.

Share

answered Oct 2, 2008 at 5:33

community wiki
Jeff Yates

Good point! c# is great for tasks involving any non-trivial amount of work, but other than that it's a pain to use. That's why i learned a scripting language :) – RCIX Dec 1, 2009 at 22:56

3

votes

If Microsoft had to invent yet another C++-like language in C# why didn't they correct Java's mistake and implement support for RAII?

Share

answered Oct 2, 2008 at 7:30

community wiki
David Holm

1 RAII depends on having deterministic destructors, which aren't feasible in a garbage-collected language. – Ferruccio Oct 2, 2008 at 14:54

RAII just requires an interface with language support. The interface (call it Resource) has one method void Release(). The language support would be a variable keyword "acquired File foo = new File()". When an acquired variable leaves scope, Release is automagically called. Simple. – jmucchiello Feb 23, 2009 at 6:50

6 That sounds like IDisposable to me. – David Dec 1, 2009 at 23:12

2 @jmucchiello: I'm afraid it's not that simple (otherwise Java and C# probably would do it :-)). The problem is that the instance referenced by a variable can survive the instance (if the instance is copied & stored somewhere, such as in a collection). To clean up theses instances, you need garbage collection, which cannot support RAII.. – sleske Dec 2, 2009 at 0:40

Certainly full support for RAII would require having types which neither inherit from Object nor allow copying except via designated methods. I'm not sure why that shouldn't be possible, though. It peeves me that neither vb.net nor C# provide a convenient means for auto-generating a Dispose method to clean up all specially-tagged IDisposable fields, nor a means of having a cleanup method get called when a constructor throws an exception after an object is partially created. – supercat Aug 29, 2011 at 2:49

3 votes

Case sensitivity.

What kinda hangover do you need to think that differentiating two identifiers solely by caSE is a great idea?

Share

answered Oct 2, 2008 at 7:42

community wiki
Yuvi

7 In Java it's quite conventional to name your classes with a capital first, and your instances in lower case. It is very common to see something like: Person person = new Person(); – Dean Rather Oct 2, 2008 at 13:48

3 It's seems every language designer except for Bill Gates was hungover. – Vasil Oct 3, 2008 at 7:40

7 I totally have the opposite view! I like how in Haskell, case sensitivity is used sort of as part of the syntax (types and constructors always uppercase, variables always lower). I find it to be extremely elegant. – Jared Updike Oct 29, 2008 at 19:01

I think case sensitivity is great in statically-typed languages, but for dynamic ones where there is no compiler to catch your errors, it can get really annoying. – Edan Maor Dec 1, 2009 at 23:13

3 votes

I hate semi-colons. I find they add a lot of noise and you rarely need to put two statements on a line. I prefer the style of Python and other languages... end of line is end of a statement.

Share

answered Feb 22, 2009 at 20:45

community wiki
cdmckay

I'd say they're more useful the other way round - when the end of the line is *not* supposed to be the end of the statement, so you can break up long lines. – Benno Dec 1, 2009 at 23:17

I used to dislike semicolons, but then I saw VB which does the opposite. YUCK! – Christian Hayter Dec 1, 2009 at 23:22

3 votes

Any language that can't fully decide if Arrays/Loop/string character indexes are zero based or one based.

I personally prefer zero based, but any language that mixes the two, or lets you "configure" which is used can drive you bonkers. (Apache Velocity - I'm looking in your direction!)

snip from the VTL reference (default is 1, but you can set it to 0):

```
# Default starting value of the loop
# counter variable reference.
directive.foreach.counter.initial.value = 1
```

(try merging 2 projects that used different counter schemes - ugh!)

Share

answered Feb 28, 2009 at 17:12

community wiki
scunliffe

3
votes

In no particular order...

OCaml

- Tuples definitions use `*` to separate items rather than `,`. So, `("Juliet", 23, true)` has the type `(string * int * bool)`.
- For being such an awesome language, the documentation has this haunting comment on threads: "*The threads library is implemented by time-sharing on a single processor. It will not take advantage of multi-processor machines. Using this library will therefore never make programs run faster.*" JoCaml doesn't fix this problem.
- ^^^ I've heard the Jane Street guys were working to add concurrent GC and multi-core threads to OCaml, but I don't know how successful they've been. I can't imagine a language without multi-core threads and GC surviving very long.
- No easy way to explore modules in the toplevel. Sure, you can write `module q = List;;` and the toplevel will happily print out the module definition, but that just seems hacky.

C#

- Lousy type inference. Beyond the most trivial expressions, I have to give types to generic functions.
- All the LINQ code I ever read uses method syntax, `x.Where(item => ...).OrderBy(item => ...)`. No one ever uses expression syntax, `from item in x where ... orderby ... select`. Between you and me, I think expression syntax is silly, if for no other reason than that it looks "foreign" against the backdrop of all other C# and VB.NET code.

LINQ

Every other language uses the industry standard names are Map, Fold/Reduce/Inject, and Filter. LINQ has to be different and uses Select, Aggregate, and Where.

Functional Programming

Monads are *mystifying*. Having seen the Parser monad, Maybe monad, State, and List monads, I can understand perfectly how the code works; however, as a general design pattern, I can't seem to look at problems and say "hey, I bet a monad would fit perfect here".

Ruby

GRRRRAAAAAAAH!!!!! I mean... seriously.

VB

```
Module Hangups
    Dim _juliet as String = "Too Wordy!"

    Public Property Juliet() as String
        Get
            Return _juliet
        End Get
        Set (ByVal value as String)
            _juliet = value
        End Set
    End Property
End Module
```

And setter declarations are the bane of my existence. Alright, so I change the data type of my property -- now I need to change the data type in my setter too? Why doesn't VB borrow from C# and simply incorporate an implicit variable called value?

.NET Framework

I personally like Java casing convention: classes are PascalCase, methods and properties are camelCase.

Share

edited Dec 2, 2009 at 18:37

community wiki
2 revs
Juliet

The vb property declarations are a peeve of mine too, especially given that a very common pattern is to implement read-only public or protected properties with a private backing field. – supercat Aug 29, 2011 at 2:41

2
votes

In C/C++, it annoys me how there are different ways of writing the same code.

e.g.

```
if (condition)
{
   callSomeConditionalMethod();
}
callSomeOtherMethod();
```

vs.

```
if (condition)
   callSomeConditionalMethod();
callSomeOtherMethod();
```

equate to the same thing, but different people have different styles. I wish the original standard was more strict about *making a decision* about this, so we wouldn't have this ambiguity. It leads to arguments and disagreements in code reviews!

Share

edited Oct 2, 2008 at 5:52

community wiki
2 revs
LeopardSkinPillBoxHat

Languages aren't designed to prevent arguments, they're designed to provide abstractions from machine level code in order to tell a computer what to do. If you make the standard strict in one situation, you'll get arguments elsewhere. The example you give is a corner-case for single line statements. – Jeff Yates Oct 2, 2008 at 5:34

Yes, I agree that this was a corner-case situation. It was intended to be a single example of my gripe ("In C/C++, there are different ways of writing the same code"). It was not my sole problem. – LeopardSkinPillBoxHat Oct 2, 2008 at 5:52

6 "In C/C++, there are different ways of writing the same code" Man, did you ever program in Perl? =) – Seiti Nov 3, 2008 at 1:53

Don't argue about it in code reviews. Let the people who care do the work. Don't do any formatting work yourself, be productive instead, and then *you* will get a raise ;-) – Jonas Kölker Feb 22, 2009 at 21:54

2 votes

I found Perl's use of "defined" and "undefined" values to be so useful that I have trouble using scripting languages without it.

Perl:

```
($lastname, $firstname, $rest) = split(' ',
$fullname);
```

This statement performs well no matter how many words are in $fullname. Try it in Python, and it explodes if $fullname doesn't contain exactly three words.

Share

answered Oct 2, 2008 at 6:32

community wiki
Barry Brown

1 Yeah, this is a bit of a PITA in Python. I usually do something like: fields = fullname.split(); fields += [None]*(len(fields)-3); lastname, firstname, rest = fields . Definitely not as convenient. – Dan Lenski Oct 2, 2008 at 6:38

Exactly! Do What I Want, darn it! :) – Barry Brown Oct 2, 2008 at 6:44

2
votes

SQL, they say you should not use cursors and when you do, you really understand why...
its so heavy going!

```
    DECLARE mycurse CURSOR LOCAL FAST_FORWARD
READ_ONLY
        FOR
        SELECT field1, field2, fieldN FROM atable

    OPEN mycurse
    FETCH NEXT FROM mycurse INTO @Var1, @Var2,
@VarN

    WHILE @@fetch_status = 0
    BEGIN
```

```
        -- do something really clever...

        FETCH NEXT FROM mycurse INTO @Var1, @Var2, @VarN
    END
    CLOSE mycurse
    DEALLOCATE mycurse
```

Share

answered Oct 2, 2008 at 7:32

community wiki
ThatBloke

2
votes

Although I program primarily in python, It irks me endlessly that lambda body's must be expressions.

I'm still wrapping my brain around JavaScript, and as a whole, Its mostly acceptable. Why is it so hard to create a namespace. In TCL they're just ugly, but in JavaScript, it's actually a rigmarole AND completely unreadable.

In SQL how come everything is just one, huge freekin SELECT statement.

Share

answered Feb 22, 2009 at 20:28

community wiki
SingleNegationElimination

I have to 1-up you on the comment about Python `lambda` s - its interesting Guido butchered such a useful LISP feature. The usability wasn't entirely killed, but a lot of the flexibility was removed in making it an expression. – new123456 Apr 22, 2011 at 1:33

2

votes

In Ruby, I very strongly dislike how methods do not require `self.` to be called on current instance, but properties do (otherwise they will clash with locals); i.e.:

```
def foo()
   123
end

def foo=(x)
end

def bar()
    x = foo() # okay, same as self.foo()
    x = foo   # not okay, reads unassigned local
variable foo
    foo = 123 # not okay, assigns local variable
foo
end
```

To my mind, it's very inconsistent. I'd rather prefer to either always require `self.` in all cases, or to have a sigil for locals.

Share

answered Dec 1, 2009 at 23:26

community wiki

Pavel Minaev

1
vote

Java's packages. I find them complex, more so because I am not a corporation. I vastly prefer namespaces. I'll get over it, of course - I'm playing with the Android SDK, and Eclipse removes a lot of the pain. I've never had a machine that could run it interactively before, and now I do I'm very impressed.

Share

answered Oct 2, 2008 at 6:38

community wiki
Bernard

How much of a difference is there between packages and namespaces? – Zifre Mar 29, 2009 at 16:14

1 Well, conceptually, not much. What I was commenting on was the way packages rely on filesystem structure, and the way you're supposed to base it off your domain (com.google.Whatever) instead of namespace MyProject { whatever }. – Bernard Mar 29, 2009 at 23:03

Ok, I guess I agree with you there on the filesystem structure thing, but at least com.google.Whatever helps prevent conflicts. – Zifre Apr 5, 2009 at 15:23

Well, basing package names off your domain name is just a convention and not enforced in any way. It's still a good idea, because it avoids name clashes. Also, I personally find the parallelism between package and fs structure rather useful, as it makes classes easier to find. But that might be debatable... – sleske Dec 2, 2009 at 0:38

I will admit that the package/fs parallelism has grown on me, somewhat. – Bernard Dec 2, 2009 at 1:40

1

vote

Prolog's `if-then-else` syntax.

```
x -> y ; z
```

The problem is that "`;`" is the "`or`" operator, so the above looks like "`x` implies `y` or `z`".

Share

answered Oct 6, 2008 at 1:34

community wiki
Hugh Allen

Wow, that is indeed inscrutable! – Dan Lenski Oct 6, 2008 at 17:20

You can do the same in most scriptic languages. JS: `var a = b || 12;` will assign to a b if it has truthful value, otherwise 12. – Radagast the Brown Mar 11, 2011 at 22:04

@Radagast the Brown: the above `->` operator is not assignment. It is implication (like `if` ... `then`). `x` is the condition, `y` and `z` are the true and false branches. – Hugh Allen Mar 26, 2011 at 5:05

1

vote

Java

- Generics (Java version of templates) are limited. I can not call methods of the class and I can not create instances of the class. Generics are used by containers, but I can use containers of instances of Object.
- No multiple inheritance. If a multiple inheritance use does not lead to diamond problem, it should be allowed. It should allow to write a default implementation of interface methods, a example of problem: the interface MouseListener has 5 methods, one for each event. If I want to handle just one of them, I have to implement the 4 other methods as an empty method.
- It does not allow to choose to manually manage memory of some objects.
- Java API uses complex combination of classes to do simple tasks. Example, if I want to read from a file, I have to use many classes (FileReader, FileInputStream).

Python

- Indentation is part of syntax, I prefer to use the word "end" to indicate end of block and the word "pass" would not be needed.
- In classes, the word "self" should not be needed as argument of functions.

C++

- Headers are the worst problem. I have to list the functions in a header file and implement them in a cpp file. It can not hide dependencies of a class. If a class A uses the class B privately as a field, if I include the header of A, the header of B will be included too.

- Strings and arrays came from C, they do not provide a length field. It is difficult to control if std::string and std::vector will use stack or heap. I have to use pointers with std::string and std::vector if I want to use assignment, pass as argument to a function or return it, because its "=" operator will copy entire structure.
- I can not control the constructor and destructor. It is difficult to create an array of objects without a default constructor or choose what constructor to use with if and switch statements.

Share

answered Apr 15, 2011 at 23:52

community wiki
Squall

0
votes

In most languages, file access. VB.NET is the only language so far where file access makes any sense to me. I do not understand why if I want to check if a file exists, I should use File.exists("") or something similar instead of creating a file object (actually FileInfo in VB.NET) and asking if it exists. And then if I want to open it, I ask it to open: (assuming a FileInfo object called fi) fi.OpenRead, for example. Returns a stream. Nice. Exactly what I wanted. If I want to move a file, fi.MoveTo. I can also do fi.CopyTo.

What is this nonsense about not making files full-fledged objects in most languages? Also, if I want to iterate through the files in a directory, I can just create the directory object and call .GetFiles. Or I can do .GetDirectories, and I get a whole new set of DirectoryInfo objects to play with.

Admittedly, Java has some of this file stuff, but this nonsense of having to have a whole object to tell it how to list files is just silly.

Also, I hate ::, ->, => and all other multi-character operators except for <= and >= (and maybe -- and ++).

Share

answered Feb 28, 2009 at 5:55

community wiki
Instance Hunter

I love =>! can you do this in VB? `myEvent += (sender, args) => {MessageBox.Show("it worked!")};` – RCIX Dec 1, 2009 at 23:01

`AddHandler myEvent, Function() MessageBox.Show("it worked!")` or something like that - don't have the IDE in front of me to check. – Christian Hayter Dec 1, 2009 at 23:26

0
votes

[Disclaimer: i only have a passing familiarity with VB, so take my comments with a grain of salt]

I Hate How Every Keyword In VB Is Capitalized Like This. I saw a blog post the other week (month?) about someone who tried writing VB code without any capital letters (they did something to a compiler that would let them compile VB code like that), and the language looked much nicer!

Share

answered Dec 1, 2009 at 23:06

community wiki
RCIX

1 Visual Basic is not case sensitive. You can write VB code all in lowercase letter if you want, and it will compile fine. But the IDE will change the case behind your back! Initial uppercase for keywords, and making variable names with same capitalization than the declaration. That's an IDE thing, not a compiler one. – PhiLho Mar 15, 2011 at 20:37

@PhiLho: I was absolutely sure that they mentioned that whatever VB compiler they were using required capitalization of keywords, and that they had to recompile a version that didn't need that. – RCIX Apr 4, 2011 at 21:26

honestly I don't know, I never used VB outside of the Visual environment doing what I describe, beyond v.6. So perhaps you are right and the IDE is changing case of keywords to fit the requirement... Wikipedia says "generally not case sensitive" (and add more precision) so we are both half right... :-) (not on the same half!) – PhiLho Apr 9, 2011 at 7:29

0
votes

My big hangup is MATLAB's syntax. I use it, and there are things I like about it, but it has so many annoying quirks.

Let's see.

- Matrices are indexed with parentheses. So if you see something like Image(350,260), you have no clue from that whether we're getting an element from the Image matrix, or if we're calling some function called Image and passing arguments to it.
- Scope is insane. I seem to recall that for loop index variables stay in scope after the loop ends.
- If you forget to stick a semicolon after an assignment, the value will be dumped to standard output.
- You may have one function per file. This proves to be very annoying for organizing one's work.

I'm sure I could come up with more if I thought about it.

Share

edited Dec 2, 2009 at 0:51

community wiki
2 revs, 2 users 93%
BigBeagle

1 2 Next