Is a variable named i unacceptable? [closed]

Asked 16 years, 3 months ago Modified 12 years, 7 months ago

Viewed 2k times

16

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 9 years ago.

Improve this question

As far as variable naming conventions go, should iterators be named `i` or something more semantic like `count`? If you don't use `i`, why not? If you feel that `i` is acceptable, are there cases of iteration where it shouldn't be used?

variables coding-style iterator naming-conventions

Share

Improve this question

edited May 5, 2012 at 19:40

Facundo Casco

10.5k ● 8 ● 44 ● 65

Follow

asked Sep 25, 2008 at 0:24

VirtuosiMedia

53.3k ● 21 ● 98 ● 140

Take a look at <stackoverflow.com/questions/101070/…>, which addresses this question. – Jon Ericson Sep 25, 2008 at 0:27

6 Only in team projects. Because there is no i in team! – Sedat Kapanoglu Nov 3, 2010 at 17:49

21 Answers

Sorted by: Highest score (default)

35

Depends on the context I suppose. If you where looping through a set of Objects in some collection then it should be fairly obvious from the context what you are doing.

```
for(int i = 0; i < 10; i++)
{
    // i is well known here to be the index
    objectCollection[i].SomeProperty = someValue;
}
```

However if it is not immediately clear from the context what it is you are doing, or if you are making modifications to the index you should use a variable name that is more indicative of the usage.

```
for(int currentRow = 0; currentRow < numRows;
currentRow++)
{
```

```
    for(int currentCol = 0; currentCol < numCols;
currentCol++)
    {
        someTable[currentRow][currentCol] =
someValue;
    }
}
```

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:32

Josh

44.9k ● 7 ● 104 ● 124

I would use iRow and iCol, but then I started on FORTRAN – Martin Beckett Sep 25, 2008 at 0:57

FORTRAN is still awesome for scientific computing. Lol, scientists will actually laugh at you if you mention other languages. – Josh Sep 25, 2008 at 0:59

Double any single character variable to make searching useful. ii v. i, xx v. x. – user7116 Sep 25, 2008 at 1:02

The 'double any single character' trick is less useful when you have an IDE with built in refactoring wizards. As for Josh's answer, this is basically a perfect example of what to do. Respeck! – Tim Frey Sep 25, 2008 at 1:04

3 currentRow and currentCol could easily be called just row and col. – mat_geek Sep 25, 2008 at 3:00

"i" *means* "loop counter" to a programmer. There's nothing wrong with it.

14

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:30

Andy Lester

93.5k ● 15 ● 104 ● 159

Here's another example of something that's perfectly okay:

6

```
foreach (Product p in ProductList)
{
    // Do something with p
}
```

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:31

Ian P

13k ● 6 ● 50 ● 70

3 Absolutely. Nobody ever complains that mathematical proofs which read, "consider a continuous function, f" are incomprehensible because the function is called "f" rather than "a_continuous_function". – Steve Jessop Sep 25, 2008 at 0:43

I tend to use i, j, k for very localized loops (only exist for a short period in terms of number of source lines). For

5

variables that exist over a larger source area, I tend to use more detailed names so I can see what they're for without searching back in the code.

By the way, I think that the naming convention for these came from the early Fortran language where I was the first integer variable (A - H were floats)?

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:31

paxdiablo

880k ● 241 ● 1.6k ● 2k

That's an interesting little tidbit about the history. Thanks.
– VirtuosiMedia Sep 25, 2008 at 0:32

1 as they say "God is real, unless declared integer"
– Greg Rogers Sep 25, 2008 at 0:36

3

`i` is acceptable, for certain. However, I learned a tremendous amount one semester from a C++ teacher I had who refused code that did not have a descriptive name for every single variable. The simple act of naming everything descriptively forced me to think harder about my code, and I wrote better programs after that course, not from learning C++, but from learning to name everything. *Code Complete* has some good words on this same topic.

Share Improve this answer

Follow

edited Sep 25, 2008 at 0:47

answered Sep 25, 2008 at 0:33

Clinton Dreisbach

7,880 ● 6 ● 31 ● 27

3

i is fine, but something like this is not:

```
for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        string s = datarow[i][j].ToString(); // or
worse
    }
}
```

Very common for programmers to inadvertently swap the i and the j in the code, especially if they have bad eyesight or their Windows theme is "hotdog". This is always a "code smell" for me - it's kind of rare when this doesn't get screwed up.

Share Improve this answer

Follow

answered Sep 25, 2008 at 1:43

MusiGenesis

75.3k ● 41 ● 197 ● 338

I never really thought much about why, but I tend to use j and k for indexes into matrices, but I suppose it is possible to get i and j confused if you have poor eyesight. – Cercerilla Nov 3, 2010 at 17:56

3

i is so common that it is acceptable, even for people that love descriptive variable names.

What is absolutely unacceptable (and a sin in my book) is using i,j, or k in any other context than as an integer index in a loop.... e.g.

```
foreach(Input i in inputs)
{
    Process(i);

}
```

Share Improve this answer

Follow

answered Sep 25, 2008 at 2:00

Giovanni Galbo

13.1k ● 13 ● 61 ● 79

2

i is definitely acceptable. Not sure what kind of justification I need to make -- but I do use it all of the time, and other very respected programmers do as well.

Social validation, I guess :)

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:26

Ian P

13k ● 6 ● 50 ● 70

2

Yes, in fact it's preferred since any programmer reading your code will understand that it's simply an iterator.

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:27

Ben Hoffstein
103k ● 8 ● 106 ● 121

2

What is the value of using i instead of a more specific variable name? To save 1 second or 10 seconds or maybe, maybe, even 30 seconds of thinking and typing?

What is the cost of using i? Maybe nothing. Maybe the code is so simple that using i is fine. But maybe, maybe, using i will force developers who come to this code in the future to have to think for a moment "what does i mean here?" They will have to think: "is it an index, a count, an offset, a flag?" They will have to think: "is this change safe, is it correct, will I be off by 1?"

Using i saves time and intellectual effort when writing code but may end up costing more intellectual effort in the future, or perhaps even result in the inadvertent introduction of defects due to misunderstanding the code.

Generally speaking, most software development is maintenance and extension, so the amount of time spent reading your code will vastly exceed the amount of time spent writing it.

It's very easy to develop the habit of using meaningful names everywhere, and once you have that habit it takes only a few seconds more to write code with meaningful

names, but then you have code which is easier to read, easier to understand, and more obviously correct.

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:36

Wedge

19.8k ● 7 ● 50 ● 71

2

I use i for short loops.

The reason it's OK is that I find it utterly implausible that someone could see a declaration of iterator type, with initializer, and then three lines later claim that it's not clear what the variable represents. They're just pretending, because they've decided that "meaningful variable names" must mean "long variable names".

The reason I actually do it, is that I find that using something unrelated to the specific task at hand, and that I would only ever use in a small scope, saves me worrying that I might use a name that's misleading, or ambiguous, or will some day be useful for something else in the larger scope. The reason it's "i" rather than "q" or "count" is just convention borrowed from mathematics.

I don't use i if:

- The loop body is not small, or
- the iterator does anything other than advance (or retreat) from the start of a range to the finish of the loop:

i doesn't necessarily have to go in increments of 1 so long as the increment is consistent and clear, and of course might stop before the end of the iterand, but if it ever changes direction, or is unmodified by an iteration of the loop (including the devilish use of iterator.insertAfter() in a forward loop), I try to remember to use something different. This signals "this is not just a trivial loop variable, hence this may not be a trivial loop".

Share Improve this answer Follow

edited Sep 25, 2008 at 1:01

answered Sep 25, 2008 at 0:37

Steve Jessop

279k ● 40 ● 469 ● 709

1

If the "something more semantic" is "iterator" then there is no reason not to use i; it is a well understood idiom.

Share Improve this answer Follow

answered Sep 25, 2008 at 0:29

Captain Segfault

1,726 ● 8 ● 12

1

i think i is completely acceptable in for-loop situations. i have always found this to be pretty standard and never really run into interpretation issues when i is used in this instance. foreach-loops get a little trickier and i think

really depends on your situation. i rarely if ever use i in foreach, only in for loops, as i find i to be too un-descriptive in these cases. for foreach i try to use an abbreviation of the object type being looped. e.g:

```
foreach(DataRow dr in datatable.Rows)
{
    //do stuff to/with datarow dr here
}
```

anyways, just my $0.02.

Share Improve this answer

Follow

answered Sep 25, 2008 at 11:01

pbrstreetgang

0

It helps if you name it something that describes what it is looping through. But I usually just use i.

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:26

Matthew Rapati

5,686 ● 4 ● 30 ● 48

0

As long as you are either using i to count loops, or part of an index that goes from 0 (or 1 depending on PL) to n, then I would say i is fine.

Otherwise its probably easy to name i something meaningful it its more than just an index.

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:28

Pyrolistical

28k ● 21 ● 84 ● 109

0

I should point out that i and j are also mathematical notation for matrix indices. And usually, you're looping over an array. So it makes sense.

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:32

Jason Baker

198k ● 138 ● 382 ● 520

0

As long as you're using it temporarily inside a simple loop and it's obvious what you're doing, sure. That said, is there no other short word you can use instead?

`i` is widely known as a loop iterator, so you're actually more likely to confuse maintenance programmers if you use it outside of a loop, but if you use something more descriptive (like `filecounter`), it makes code nicer.

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:32

Dan Udey

2,977 ● 2 ● 21 ● 18

0

It depends. If you're iterating over some particular set of data then I think it makes more sense to use a descriptive name. (eg. `filecounter` as Dan suggested).

However, if you're performing an arbitrary loop then `i` is acceptable. As one work mate described it to me - `i` is a convention that means "this variable is only ever modified by the `for` loop construct. If that's not true, don't use `i`"

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:38

Andrew Edgecombe

40.3k ● 3 ● 38 ● 63

0

The use of i, j, k for INTEGER loop counters goes back to the early days of FORTRAN.
Personally I don't have a problem with them so long as they are INTEGER counts.
But then I grew up on FORTRAN!

Share Improve this answer

Follow

answered Sep 25, 2008 at 0:39

DaveF

6,829 ● 6 ● 27 ● 20

---

my feeling is that the *concept* of using a single letter is fine for "simple" loops, however, i learned to use double-

0

letters a long time ago and it has worked out great.

i asked a similar question last week and the following is part of my own answer:

```
// recommended style                    •    //
"typical" single-letter style
                                        •
for (ii=0; ii<10; ++ii) {               •    for (i=0;
i<10; ++i) {
    for (jj=0; jj<10; ++jj) {           •        for
(j=0; j<10; ++j) {
        mm[ii][jj] = ii * jj;           •
m[i][j] = i * j;
    }                                   •        }
}                                       •    }
```

in case the benefit isn't immediately obvious: searching through code for any single letter will find many things that *aren't* what you're looking for. the letter `i` occurs quite often in code where it isn't the variable you're looking for.

i've been doing it this way for at least 10 years.

note that plenty of people commented that either/both of the above are "ugly"...

Share Improve this answer

Follow

edited May 23, 2017 at 12:25

Community Bot

1 ● 1

answered Sep 25, 2008 at 16:46

just mike

1,172 ● 3 ● 12 ● 22

1 And for people with editors that can search on word breaks, completely pointless. – Jon Ericson Sep 25, 2008 at 16:48

Offtopic a little : why do you preincrement the loop variable in the for statement? I usually see **POSTINCREMENT** instead of preincrement. – Andrei Rînea Oct 4, 2008 at 16:07

efficiency. it comes from my early experience with the C language when the assembly optimizers weren't perfect. if you post-increment, the expressions is first evaluated, then incremented, then evaluated again. if you pre-increment, there's no initial evaluation, just incremented then evaluated. – just mike Oct 4, 2008 at 22:03

-1

I am going to go against the grain and say no.

For the crowd that says "i is understood as an iterator", that may be true, but to me that is the equivalent of comments like 'Assign the value 5 to variable Y. Variable names like comment should explain the why/what not the how.

To use an example from a previous answer:

```
for(int i = 0; i < 10; i++)
{
```

```
    // i is well known here to be the index
    objectCollection[i].SomeProperty = someValue;
}
```

Is it that much harder to just use a meaningful name like so?

```
for(int objectCollectionIndex = 0;
objectCollectionIndex  < 10; objectCollectionIndex
++)
{

objectCollection[objectCollectionIndex].SomeProperty
= someValue;
}
```

Granted the (borrowed) variable name objectCollection is pretty badly named too.

Share Improve this answer

Follow

answered Nov 3, 2010 at 17:44

JohnFx

34.9k ● 18 ● 107 ● 166