

Ruby - Feature #17496

Add constant Math::TAU

12/31/2020 07:27 PM - jzakiya (Jabari Zakiya)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		
Description Excuse me if this has been brought up before. This is a simple proposal to add the math constant <code>Math::TAU = 2*Math::PI</code> . See: https://tauday.com/ TAU has been included in a growing number of languages (Rust, Python3, Julia, et al), and this would add Ruby to that list, and make it even cooler. :-)		
Related issues: Related to Ruby - Feature #20404: <code>`2pi`</code> Rejected Is duplicate of Ruby - Feature #4897: Define <code>Math::TAU</code> and <code>BigMath.TAU</code> . The "... Rejected		

History

#1 - 12/31/2020 09:13 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Rejected

jzakiya (Jabari Zakiya) wrote:

Excuse me if this has been brought up before.

It has been brought up before, by you in [#13694](#), which was itself a duplicate of [#4897](#). I'm rejecting this as a duplicate. However, if you think circumstances have changed since the last rejection, please post what has changed since then.

#2 - 12/31/2020 10:16 PM - jzakiya (Jabari Zakiya)

Well, I guess this regresses to a philosophical|religious issue, it's obviously not technical. :-)

But how can adding TAU in anyway hurt Ruby?

A short list of languages I know of which now include TAU:

D
C# (.NET)
F#
Nim
Zig
Rust
Julia
Python3

What has changed is TAU is gaining more traction in math|science, and is being included in more languages.

But really, Ruby 3.0 has added so much stuff to it, some purely for looks (endless methods), I don't understand the resistance to adding this one constant, as it requires absolutely no change to the language.

If programmer happiness is really a high priority, this will make allot of us very happy.

#3 - 12/31/2020 10:18 PM - chrisseaton (Chris Seaton)

Relevant context is probably Matz's reason for rejection and criteria for reconsidering.

<https://bugs.ruby-lang.org/issues/4897#note-41>

#4 - 12/31/2020 10:47 PM - jzakiya (Jabari Zakiya)

From that post 3 years ago.

For the time being, use math-tau gem.

I will add it after (and only after) it became time-proven major, probably due to Python's tau.

Matz.

Well, Python3 added it, so when do you (empirically?, subjectively?) establish it's become time-proven?

How many other language adoptions does it take (oh btw, added Zig to the list too)? . :-(

#5 - 01/01/2021 02:13 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #4897: Define Math::TAU and BigMath.TAU. The "true" circle constant, $\text{Tau}=2\pi$. See <http://tauday.com/> added

#6 - 01/01/2021 02:14 AM - mame (Yusuke Endoh)

jzakiya (Jabari Zakiya) wrote in [#note-4](#):

Well, Python3 added it, so when do you (empirically?, subjectively?) establish it's become time-proven?

IMO, the sufficient condition is that it appears in some major textbooks for school.

#7 - 01/01/2021 02:41 AM - mrkn (Kenta Murata)

At least Julia doesn't have tau in its core now. There is Tau.jl, but it isn't a standard library. Following Julia's stance to tau, we can say that it is enough by providing a gem to define Math::TAU for now.

I'd like to know how to use tau in languages you listed up other than Julia, Python3, and C#. Could you tell me the way?

jzakiya (Jabari Zakiya) wrote in [#note-4](#):

Well, Python3 added it, so when do you (empirically?, subjectively?) establish it's become time-proven?

I think we can add TAU in core if most mathematicians usually use tau as the circle constant in their academic papers.

I don't know how the situation of tau as the circle constant is changed from the time of the previous proposal. I'd like to show my response to the previous pull-request again.

<https://github.com/ruby/ruby/pull/644#issuecomment-123082639>

- It looks to me that tau as the circle constant isn't popular enough to be introduced in Ruby's core although it has good property.
- So introducing tau is hurt the future Ruby now. Core constants cannot remove easily.
- It's not Ruby's role to help further the acceptance and adoption of tau.
- I found that Python and Julia rejected similar proposals. See <https://bugs.python.org/issue12345> and <https://github.com/JuliaLang/julia/pull/4864>.

So, I want to know how many mathematicians accept such tau. Do you know how many academic papers that uses tau as the circle constant have been published so far?

#8 - 01/01/2021 06:46 AM - jzakiya (Jabari Zakiya)

So, I want to know how many mathematicians accept such tau. Do you know how many academic papers that uses tau as the circle constant have been published so far?

How can anyone ever empirically answer these question? And why should they be "the" deciding factor?

Have you done any research to answer these questions?

Have you read the Tau Manifesto? <https://tauday.com/tau-manifesto>

Have you seen the Tau videos, like the Numberphile one? https://youtu.be/83ofi_L6eAo

Every year there is a growing body of work promoting|using Tau, and more languages including it.

I don't think I can give you anymore of an objective, rational, and logical reason than these sources (with others).

Like I said, this seems to ultimately be a philosophical|religious issue, and you're just not believers yet.

Maybe in time hopefully you will be, but you must find your own way to acceptance, because others already have.

Let's hope 2021 will be a better new year. Happy New Year!

Addition:

This video (from 2017) gives you one textbook that uses TAU, and other places of usage.

<https://www.youtube.com/watch?v=k7MuXCOIE6M&feature=youtu.be>

#9 - 01/01/2021 03:46 PM - jzakiya (Jabari Zakiya)

To help answer this question,

IMO, the sufficient condition is that it appears in some major textbooks for school.

and this,

So, I want to know how many mathematicians accept such tau. Do you know how many academic papers that uses tau as the circle constant have been published so far?

I reviewed a couple of the TAU videos by Micheal Hartl, the author of The TAU Manifesto and creator of TAUDAY. He provides examples of academic and educational resources where TAU is being used and formally taught. I have extracted some of that information here as examples to the answer to these questions.

Textbook

College Trigonometry: With Extensive Use of the Tau Transcendental (free downloadable version)

<https://studyres.com/doc/17428279/college-trigonometry-with-extensive-use-of-the-tau-transc...>

Educational videos, and learning resources

Kahn Academy

<https://www.khanacademy.org/math/trigonometry/unit-circle-trig-func/long-live-tau/v/tau-versus-pi>

<https://www.khanacademy.org/math/trigonometry/unit-circle-trig-func/long-live-tau/v/pi-is-still-wrong>

<https://blog.khanacademy.org/happy-tau-day/>

Numberphile Video

https://youtu.be/83ofi_L6eAo

Papers

PI is Wrong - <https://www.math.utah.edu/~palais/pi.html>

The TAU Manifesto - <https://tauday.com/tau-manifesto>

You can find many more examples of TAU's use with more research|searching.

It has been 11 years now since 2010 when Michael Hartl wrote the The TAU Manifesto and started TAUDAY.

It is gaining greater academic, educational, and scientific acceptance|use every year.

It is being included into more software languages and packages every year.

Hopefully, the totality of this growing body of work will establish TAU has met Matz's time-proven criteria.

#10 - 01/01/2021 03:54 PM - chrisseaton (Chris Seaton)

It's difficult to work out if this is a genuine proposal or a meme.

Surely Jabari is joking when they say 'you must find your own way to acceptance, because others already have'? Is this a mathematical argument or a sort of play on cults? A lot of the language you're using and used in the 'manifesto' also just seem like huge red flags that a joke is being played on busy software maintainers here. The manifesto itself admits it's 'quirky, quixotic', and it definitely seems more fun than serious proposal. The Python bug tracker issue seems to have lots of people who think it's just a joke.

Can you clarify if this is a serious proposal, and if so explain why it seems to have so many red flags both to people here and for example in the Python issue tracker? Or is it half serious proposal and half something whimsical as well?

Because I can't read the tone and I think others can't as well. Can you help give some frank context?

#11 - 01/01/2021 04:42 PM - jzakiya (Jabari Zakiya)

This is a very serious proposal, and TAU is a very serious concept - the unit turn.

As Michael Hartl raises, trying to get people to accept TAU as better than PI is more about emotional attachment to a familiar concept, than an objective assessment and understanding of the need to change the thinking.

For humans, change is hard, especially for things they are familiar with.

The best time to teach anything (math, music, art, swimming, dietary habits, etc) is when humans are young, and haven't developed a resistance to

change from something they are familiar with (whether good or bad). Habits are hard to break.

Look at how hard it is to get people to accept they need to wear facemasks to protect themselves|others in this pandemic!

Everybody here is smart. Everybody here has been trained in the scientific method, to be (mostly) objective and rational.

But everybody here is also human, and habits die hard.

As stated, Python went through a long process to finally come to a point where enough people accepted including TAU, so now its in Python3. GO is having a harder time accepting it. <https://github.com/golang/go/issues/40663>

But D, C#, F#, Nim, Zig, Rust, Julia, and others I just don't know about, have accepted it. Why?

The simple answer is because those communities are comprised of different people who have accepted it, that's all.

They've probably read the same materials, seen the same videos, raised the same arguments, but in the end they said yes to including it.

So when I said

you must find your own way to acceptance, because others already have'?

that's exactly what I mean. I don't know what it will take for the people who control Ruby's development to accept that TAU should be included. Only you will know when you reach that point.

#12 - 01/01/2021 05:04 PM - larskanis (Lars Kanis)

Almost 25 years ago I started my computer career with 3D programming. Very low level to that times, but using a lot of trigonometric computations. Using PI always felt like something half done. It's strange to tell the computer it should turn the object by 6 half rounds (which is PI) instead of 3 full rounds (which is TAU).

I don't remember any scientific publications using TAU and given that it's just 2π , I would never add an additional gem just for this constant. However from a practical programmer perspective TAU is often more meaningful than PI. So IMHO this is something that should be added to ruby core.

#13 - 01/01/2021 08:58 PM - jzakiya (Jabari Zakiya)

FYI.

Yesterday (the last day of 2020) before I created this issue to include TAU in Ruby, literally seconds before, I opened the exact same request in Crystal, as Ruby|Crystal are my two favorite languages.

After one day, Crystal is going to add it.

<https://github.com/crystal-lang/crystal/issues/10165>

This is what I mean by it's *just a matter of acceptance*.

In the case of Crystal a main dev read the Manifesto (which he never heard of before) and immediately accepted|embraced the concept, and even wants to promote it via the language. Here, it's been met with caution and resistance. Same issue, same information, different people reacting to it differently.

#14 - 01/01/2021 09:03 PM - chrisseaton (Chris Seaton)

I don't think this manifesto helps itself by conflating a serious argument with a lot of silliness and jokes. You can see how people react to it by wondering if they're being trolled.

#15 - 01/01/2021 09:27 PM - jzakiya (Jabari Zakiya)

I don't think this manifesto helps itself by conflating a serious argument with a lot of silliness and jokes. You can see how people react to it by wondering if they're being trolled.

Well, I don't see anything specific in the Manifesto that is silly or a joke. At the end he answers actual questions that have been posed to him that you may take as silly questions, but that doesn't detract from the merits of the issue. And in parts of some of his videos he takes a lighthearted approach to redefining the culture around PI, but that's to get people to loosen up their resistances.

Remember, most people in the field of physics didn't just jump on the Relativity bandwagon immediately, from that (then) nobody Einstein. These things take time for each individual to come to peace in their own minds about the need to change their thinking. Look at Ruby. Until Rails came along it was *just-another-language* that most people never heard of|used.

So marketing is a big part of getting people to accept ideas. And if TAUDAY, and TAU T-shirts, and videos about two pies, make it more comfortable for people to understand|embrace the concept, then consider it effective silliness.

I would just urge you not to be blind to the overwhelming mathematical, scientific, and pedagogical benefits TAU brings, and try to understand why other people do embrace it, and actively use it.

And since that is the case, then including it will make Ruby more relevant to them, and bring them more happiness.

#16 - 01/01/2021 09:30 PM - chrisseaton (Chris Seaton)

Ok well this issue is currently marked rejected. You need to convince someone to change that first. Then you need to add it to the list of issues to be considered by Matz this month <https://bugs.ruby-lang.org/issues/17480>.

#17 - 01/01/2021 09:41 PM - jzakiya (Jabari Zakiya)

OK, well then I request this issue be reclassified as Open, by whoever has the power to do that.

Also, as it seems Matz's last response to a similar request was over 3 years ago, I request that whoever has the power to request he review this issue again now do that too.

#18 - 01/01/2021 09:42 PM - chrisseaton (Chris Seaton)

I request that whoever has the power to request he review this issue again now do that too.

That's done by adding to the list of issues to be considered I linked - when you have this as not rejected you can add to there.

#19 - 01/01/2021 09:48 PM - jzakiya (Jabari Zakiya)

Well since I don't have the power to reOpen it, will you then see that it is?

Otherwise, this is just going around in an infinite circle, and I don't need the new year starting out like this.

Since you know what needs to be done, I am asking you (or someone else who can) to perform the mechanics to get it done.

#20 - 01/01/2021 10:12 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Rejected to Open

Personally, I don't think that anything significant has changed in the last three years this area. And the comparison of Tau vs Pi to Relativity vs Newtonian Mechanics is laughable.

However, I also don't want to be the one to stop debate on this issue, so I'm reopening it. However, I recommend to Matz that he rejects it again. I think a good time to add Math::TAU is when the majority of mathematics textbooks switch from Pi to Tau, and not before.

#21 - 01/01/2021 10:14 PM - chrisseaton (Chris Seaton)

@jzakiya you just need to comment on <https://bugs.ruby-lang.org/issues/17480> then, copying the format of the other comments.

#22 - 01/01/2021 10:34 PM - ufuk (Ufuk Kayserilioglu)

I also don't understand why Math::TAU needs to be added to Ruby, even though I am familiar with the manifesto and think it might have a merit. However, it is just a redefinition of a basic mathematical constant that doesn't bring much else to the table other than making learning some trig concepts simpler. Moreover, the manifesto explicitly states:

How can we switch from π to τ ?

The next time you write something that uses the circle constant, simply say "For convenience, we set $\tau=2\pi$ ", and then proceed as usual.

Which means that all you have to do is:

```
Math::TAU = 2 * Math::PI
```

in your codebase if you want to use it, as suggested by the manifesto. There is no need for Ruby to add all possible constants to the language, especially not the ones that are expressible in terms of constants that already exist.

Basically, it boils down to: Is there a mathematical operation you can't perform if Ruby does not define Tau? I'd say: no. Happy to hear otherwise.

#23 - 01/03/2021 10:45 PM - jzakiya (Jabari Zakiya)

you just need to comment on <https://bugs.ruby-lang.org/issues/17480> then, copying the format of the other comments.

Thanks. Done.

Personally, I don't think that anything significant has changed in the last three years this area. And the comparison of Tau vs Pi to Relativity vs Newtonian Mechanics is laughable.

As with PI, the Newtonian view of reality had an entrenched pedagogical, professional, cultural, and even nationalistic hold on people's minds and lives. Einstein's Relativity shattered that for many, purely on these basis. So "F = ma is wrong" when trying to explain the way the natural world is observed to work in all cases. And even though Relativity has been observed to work over and over again, to explain things Newtonian mechanics can't, the Newtonians still resist(ed) accepting it emotionally (as Einstein himself later emotionally resisted accepting totally Quantum Mechanics).

However, I recommend to Matz that he rejects it again. I think a good time to add Math::TAU is when the majority of mathematics textbooks switch from Pi to Tau, and not before.

Well, that's a very arbitrary and subjective criteria, don't you think? I already presented one Open Source textbook, first released in 2015, that's been in use. And the Khan Academy - <https://www.khanacademy.org/> - maybe the largest, well known educational learning platform in the world, put out its first video on teaching|promoting TAU in 2012, and you can't get anymore main stream than it.

For a better, more extensive list of uses of TAU please view the list of citations on its wikipedia page:
[https://en.wikipedia.org/wiki/Tau_\(angle\)#Tau_proposal](https://en.wikipedia.org/wiki/Tau_(angle)#Tau_proposal)

Basically, it boils down to: Is there a mathematical operation you can't perform if Ruby does not define Tau? I'd say: no. Happy to hear otherwise.

I would like to refocus the discussion to what the proposal seeks, and why.

The impetus behind the proposal is to bring Ruby in line with the other (and growing) number of software languages and packages that now include the fundamental math constant TAU as a standard resource that is available for programmers to use. It does not require anybody here to use it, or change any code they have.

Though Bob Palais's PI is wrong - <https://www.math.utah.edu/~palais/pi.pdf> - appeared in 2001, the popular promotion of its acceptance|use took off with Michael Hartl's (of Ruby on Rails Tutorial fame, <https://www.railstutorial.org/>) 2010 article The Tau Manifesto and same year creation of TAU Day, as an alternative to PI Day. (MIT now releases its acceptance notices on PI Day (March 14; 3/14) at TAU Time, 6.28 pm).

Thus since 2010, the knowledge and understanding about TAU being the better|correct circle constant has steadily increased, adopted, and used.

A short (growing) list of software languages and packages that include|provide TAU are:

D -- <https://code.dlang.org/packages/tau>; <https://forum.dlang.org/thread/mailman.1426.1309854678.14074.digitalmars-d@puremagic.com>
C#.NET -- <https://github.com/dotnet/runtime/pull/37517>; <https://docs.microsoft.com/en-us/dotnet/api/system.math.tau?view=net-5.0>
F# -- " "
Nim -- <https://nim-lang.org/docs/math.html#TAU>
Zig -- <https://ziglang.org/documentation/0.7.1/std/#std.math>
Java|Node.js -- <https://mathjs.org/docs/reference/constants.html>; <https://mathjs.org/>
Rust -- <https://doc.rust-lang.org/stable/std/f64/consts/constant.TAU.html>
Raku -- <https://docs.raku.org/language/terms>
Julia -- <https://juliapackages.com/p/tau>; <https://github.com/JuliaLang/julia/pull/4864>
Crystal -- <https://github.com/crystal-lang/crystal/issues/10165>; <https://github.com/crystal-lang/crystal/pull/10179>
Python3 -- <https://docs.python.org/3/library/math.html>; <https://www.python.org/dev/peps/pep-0628/>
Matlab -- https://www.mathworks.com/matlabcentral/fileexchange/48636-tau-the-circle-constant?s_tid=srchtitle
Google calculator -- <https://www.google.com/search?q=%CF%84%3D>

See also: https://en.wikipedia.org/wiki/User:Waldyrrious/Tau#Support_in_tools_and_programming_languages

Why

1. Since 2010, TAU's acceptance and use has increased in academia, science, and software languages and packages
2. It requires no technical changes to Ruby.
3. It makes Ruby compatible with other popular languages (like Python and Rust).
4. It makes Ruby natively easier to use for pedagogy.
5. It will better establish Ruby as a good scientific language, particularly for unfamiliar Ruby users.
6. It has met Matz's previous criteria of being time-proven.
7. It will, ultimately, be expected to be included in every general purpose language.

In the final analysis, PI is wrong and TAU is right.

#24 - 01/05/2021 03:19 AM - mrkn (Kenta Murata)

@jzakiya I guess the resources you've listed up in the comment #9 are only describing tau's advantages or pi's disadvantages. They are not the evidence that shows the tau is usually used.

Why math-tau.gem is not enough for your usecase?

#25 - 01/12/2021 05:19 AM - akr (Akira Tanaka)

I feel a constant of $\text{Math}::\text{PI} * 2$ is useful especially for $x * 360 / (\text{Math}::\text{PI} * 2)$.

We need not only $*$ but also parenthesis.

$x * 360 / \text{Math}::\text{TAU}$ is shorter than above.

However I like another name, $\text{Math}::\text{PI}2$.

#26 - 01/12/2021 05:45 AM - shyouhei (Shyouhei Urabe)

M_PI_2 is $\pi/2$ rather than 2π . I guess $\text{Math}::\text{PI}2$ is too confusing.

#27 - 01/12/2021 05:56 AM - shyouhei (Shyouhei Urabe)

jzakiya (Jabari Zakiya) wrote in [#note-23](#):

Why

1. Since 2010, TAU's acceptance and use has increased in academia, science,

Can you tell us more about this area? A few references of research papers (preferably with an URL of doi.org) would be much appreciated. As I wrote in <https://bugs.ruby-lang.org/issues/4897#note-39>, I cannot distinguish if this constant being actively receiving low marks from academic community, or is just young.

#28 - 01/12/2021 06:02 AM - akr (Akira Tanaka)

shyouhei (Shyouhei Urabe) wrote in [#note-26](#):

M_PI_2 is $\pi/2$ rather than 2π . I guess $\text{Math}::\text{PI}2$ is too confusing.

I hope that it does not cause big confusion because Ruby has no PI_2 .

#29 - 01/12/2021 06:14 AM - akr (Akira Tanaka)

akr (Akira Tanaka) wrote in [#note-28](#):

shyouhei (Shyouhei Urabe) wrote in [#note-26](#):

M_PI_2 is $\pi/2$ rather than 2π . I guess $\text{Math}::\text{PI}2$ is too confusing.

I hope that it does not cause big confusion because Ruby has no PI_2 .

$\text{Math}::\text{TWOPI}$ is yet another idea.

#30 - 01/12/2021 10:10 AM - sawa (Tsuyoshi Sawada)

akr (Akira Tanaka) wrote in [#note-25](#):

However I like another name, $\text{Math}::\text{PI}2$.

akr (Akira Tanaka) wrote in [#note-28](#):

$\text{Math}::\text{TWOPI}$ is yet another idea.

Either one halfway nullifies the original proposal. The whole point of this issue is that τ is the basic mathematical constant related to circle, (and if π is temporarily ever needed, the latter is the one that should be described in terms of the former, i.e., $\pi := \tau / 2$). Expressing τ in terms of π goes counter to this.

P.S. I have had the same idea as the τ proponents since when I was a high school student. I believe I am among the countless people, if not the majority, in this respect.

#31 - 01/12/2021 10:38 PM - universato (Yoshimine Sato)

FYI

<https://processing.org/reference/TAU.html>

Processing may not be a general purpose language, but it has TAU and TWO_PI on its core. Processing has TAU since 2013.

DOI: 10.14736/kyb-2016-6-0943

<http://www.kybernetika.cz/content/2016/6/943>

In this paper τ is used in the Gaussian density function.

#32 - 01/13/2021 05:47 AM - shyouhei (Shyouhei Urabe)

universato (Yoshimine Sato) wrote in [#note-31](#):

FYI

<https://processing.org/reference/TAU.html>

Processing may not be a general purpose language, but it has TAU and TWO_PI on its core.
Processing has TAU since 2013.

DOI: 10.14736/kyb-2016-6-0943

<http://www.kybernetika.cz/content/2016/6/943>

In this paper τ is used in the Gaussian density function.

Thank you! This is the first time I actually see the practical use of this constant.

#33 - 04/02/2024 09:30 AM - hsbt (Hiroshi SHIBATA)

- *Related to Feature #20404: `2pi` added*