

# The Pi-Search Page

Search For:

See  digits starting at position

- [How does the Pi searcher work?](#)
- [Why? Why?](#)
- [Where can I get the digits of Pi? The source code?](#)
- [Why is/isn't my favorite number in Pi?](#)
- [Updates to the page and searcher \(03/14/2005\)](#)
- [Pi Numeric Trivia Bits](#)
- Dave's other pages: [[Main](#)] [[Personal](#)]

The Pi Searcher lets you search for any string of digits (up to 120 of them) in the first **200 million digits of Pi**. You can also show any substring of Pi.

## Updates

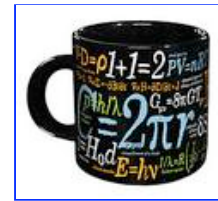
- **01/01/2010** - It's not Pi day, but 01/01/10 has a nice numerical ring to it anyway. Fixed a few long-standing Pi searcher bugs if you searched off the end of our 200M digits. Happy 2010, everybody!
- **03/14/2006** - Happy tenth birthday to the Pi Searcher, and happy pi day 2006 to everybody! This year's pi day updates: 1) New, zippy fast index-based searching; 2) The Pi searcher pages are now fully XHTML and CSS-happy, with a new look; 3) We've added a page describing [how the Pi searcher works](#). Also, a very grateful pi-thanks to our new ISP, [Xmission](#), who're a very cool group of folk to work with.
- **03/14/2005** - Happy pi day 2005. We're now up to **200 million digits**. and running on a newer, faster server. Happy pi searching!  
**New:** the [top Pi searches from 1999-2005](#).
- **01/15/2004** - fixed a silly off by one error in the pi displayer. Displays are now 1-based, just like searches.
- **08/25/2001** - You can now display any substring in Pi with the Pi Searcher. Thanks to the people who suggested this!
- **03/14/2001** (Pi Day 2001) - The pi searcher now provides **100 million** digits, made possible by the new box and the compressed searching.
- **03/13/2001** - The Pi Searcher just got a bit faster. You probably won't notice the difference, but it's about 4x faster now that it's the only computationally intensive thing running on this machine.
- **10/15/1999** - Compressed the digits two per byte, and wrote a custom search routine. The pi searcher is now nearly as fast as a linear search can be, and uses less space than before.

## Why the Pi Searcher?

In 1996, Arthur Bebak of [Netsurfer Digest](#) jokingly suggested the idea. I put the site online, linked from



[Celebrate Pi with a Pi T-Shirt](#)



[Math mug!](#)

Ads by Google

[Mobile Web Per Inquiry](#)

Over 100,000

Leads Sold

Reach 56 million

Americans!

[www.LiquidWireless.com](#)

the now-defunct Useless Web Pages Pages. The original suggestion was to find your birthday in Pi, but things got out of hand. The original pi searcher featured 1.25 million digits. It was upgraded in 1998 to 50 million, in 2001 to 100 million, and in 2005, to 200 million digits to keep up with the times. The Pi Searcher has proven both exceptionally useless (see the comments below) and occasionally useful to math & early science classes.

## How it Works

The Pi Searcher uses a combination of linear search (searching each digit one by one) for small search strings and a pre-computed index for large search strings. The result is that the Pi searcher is extremely fast -- it takes less than 1/50th of a second to handle most requests. For more information, see [how the Pi Searcher works](#).

## Finding Pi

I'm frequently asked where people can get such a ridiculously large amount of pi. Be warned that 50 million digits of pi takes up 50 megabytes. This can take up to **4 hours** to download with a 28.8k modem!

Digits	Source
Small bits of Pi	Angio.net: <a href="#">10</a> <a href="#">50</a> <a href="#">100</a> <a href="#">500</a> <a href="#">1000</a> <a href="#">10000</a> <a href="#">100000</a>
1.25 Million	<a href="#">Project Gutenberg</a>
Up to 51 billion	<a href="#">University of Tokyo</a> They are not publicly available past 4.2B. See the README file for details.
List	<a href="#">Yahoo!'s Pi Page</a>
50 million - COMPRESSED BINARY	<a href="#">angio</a> (my digits) Packed 2 digits per byte, NOT for human consumption. A hastily written unix-only decompression program is <a href="#">here</a> (requires mmap, only tested on BSD)

I've verified the digits in my 200 million against the sources listed above, and believe them to be correct. If there's an error, however, please notify me by emailing dga - at (@) - pobox DOT com and I'll upgrade.

You can find the source code to the pi searcher [here](#) in tar.gz format.

## Chances of Finding

Why can/can't I find my number in Pi? If we view Pi as a big, random string of numbers (which is close enough for our purposes), then we can figure out the odds of finding any string in the first 100 million digits of Pi:

Number Length	Chance of Finding
1-5	100%
6	Nearly 100%
7	99.995%
8	63%
9	9.5%
10	0.995%%
11	0.09995%

Happily, if you include the zeros, birthdays are 8 digits long -- so you have a 63% chance of finding your birthday in the first 100 million digits of pi. Now that we're to 200 million, the odds are up to 86%, so it'll be a while before everyone can find their birthday in Pi.

I've also posted a [more in-depth explanation of the probability of finding strings in Pi](#) for people who are curious.

## Pi Numeric Trivia Bits

---

### Self-locating Strings in Pi

(Thanks to Jeff Roulston for the impetus behind this bit of Pi trivia, and to Colin Rose for supplying the 4th string).

Pi contains a few self-locating strings, but not many. Defining self-locating depends how you count the "position". If you treat the first digit after the decimal point as digit "1" (which the pi searcher does), then you get the following numbers which can self-locate themselves in the first 100M digits of pi:

1, 16470, 44899, 79873884

If, on the other hand, you act like a computer geek and use zero based indexing, then you get these numbers:

6, 27, 13598, 43611, 24643510

### The Meaning of Life (42) and Pi

(Quoting from Scott Glazer): Trying to come up with a significant number to search for, I thought of 42 (the answer to life, the universe, and everything in Hitchhikers's Guide to the Galaxy.) 42 would be way too common of course, so I went for 424242. Came back that this shows up at position 242423. Add one (for the decimal point, I lamely rationalize here) and you get 242424, the reverse of the original input. Now that's meaningful... or something.

**[Editors Note]** Amusingly enough, the entire string returned is 242424242. If you disregard either of the ending twos, you find that it's the same position at which you find 42424242. Ahh, the palindromic possibilities inherent in a reversible meaning of life string. --Dave

### Repeating Patterns in Pi

Jonathan Day recently (02/1999) noticed that there appear to be no simple, repeating patterns longer than 10 digits. He found 9 6's at 45681781, 9 7's at 24658601 and 9 8's at 46663520. There are also the above mentioned 42's. Can you find something else?

### How many digits to find a birthday?

Many thanks to Carola Schermuly, who prompted me to figure out a most useless (but interesting) bit of Pi trivia: The maximum number of digits of Pi necessary to find any month-day combination is 60872. Interestingly enough, this is the same value with European date formats (1203 meaning March 12th) and American date formats - the same date, December 3rd, is the winner. It takes 60872 digits after the decimal point to find them.

### Loop Sequences within Pi

Dan Sikorski pointed out an interesting loop sequence within Pi. If you search for 169, it appears at position 40. If you then search for 40, it appears at position 70. Search for 70, ... and so on. The sequence Dan found is: 40, 70, 96, 180, 3664, 24717, 15492, 84198, 65489, 3725, 16974, 41702, 3788, 5757, 1958, 14609, 62892, 44745, 9385, 169, 40...

One has to wonder: What is the probability of finding a loop for any given initial search string? Or even, within the infinite expansion of pi, would all searches necessarily fall into a loop? The expected number of digits required to find a search string is proportional to the length of the string, but the requirement to loop again makes the analysis a bit tricky. Anyone know?

Doug Hafen points out that not all numbers will loop because of the self-locating strings. It's also possible to drop into a self-locating string, e.g., by searching for 211 -> 93 -> 14 -> 1. No loop. Thanks, Doug!

[Information about query privacy.](#)

---

[\[top\]](#) [\[research\]](#) [\[personal\]](#) [\[consulting\]](#)

*Last updated: Mon Mar 15 14:03:26 GMT 2010* [\[validate xhtml\]](#)  
dga - at - pobox dot com.