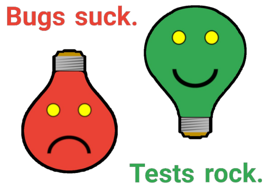


How Google Uses Bathroom Breaks to Improve Developer Knowledge and Productivity

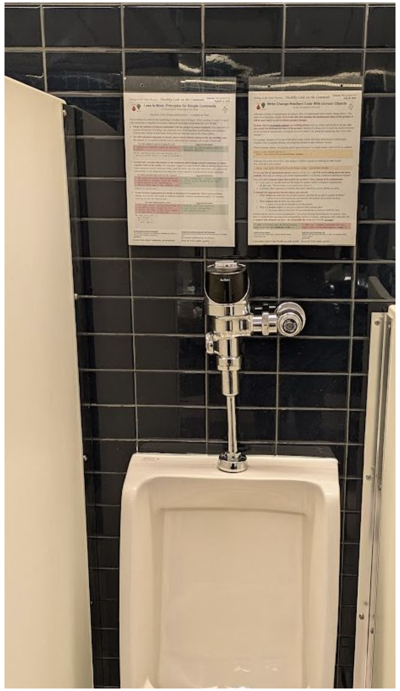
DPE Summit 2024

Andrew Trenk - Software Engineer at Google
Kanu Tewary - Technical Program Manager at Google

Google

Tech on the Toilet (TotT)

- Weekly one-page publication
- Posted in bathrooms in Google offices worldwide
- Contains actionable tips about software development



“The number one (and number two) source for software development knowledge in Google's bathrooms.”

TotT covers any topics relevant to software development

Examples:

- Coding best practices
- Unit testing
- Integration testing
- Code reviews
- Internal developer tools
- Production
- Machine learning
- Web development

Example TotT: *Reduce Code Complexity by Reducing Nesting*

With nested `if` statements:

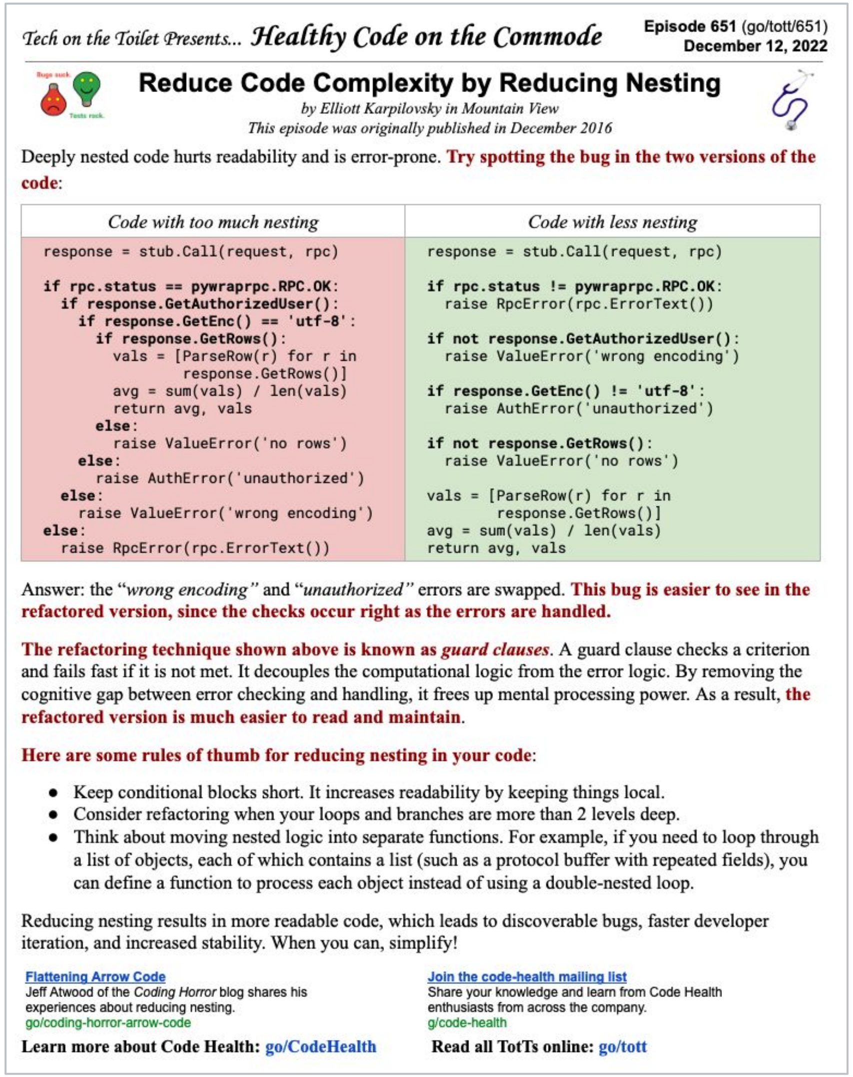
```
if response.GetAuthorizedUser():
  if response.GetEnc() == 'utf-8':
    ...
  else:
    raise AuthError('unauthorized')
else:
  raise ValueError('wrong encoding')
```

Without nested `if` statements:

```
if not response.GetAuthorizedUser():
  raise ValueError('wrong encoding')

if response.GetEnc() != 'utf-8':
  raise AuthError('unauthorized')
```

[Read it online](testing.googleblog.com) (testing.googleblog.com)



Tech on the Toilet Presents... Healthy Code on the Commode

Episode 651 (go/tott/651)
December 12, 2022

Reduce Code Complexity by Reducing Nesting

by Elliott Karpilovsky in Mountain View
This episode was originally published in December 2016

Deeply nested code hurts readability and is error-prone. **Try spotting the bug in the two versions of the code:**

Code with too much nesting

```
response = stub.Call(request, rpc)

if rpc.status == pywraprpc.RPC.OK:
  if response.GetAuthorizedUser():
    if response.GetEnc() == 'utf-8':
      if response.GetRows():
        vals = [ParseRow(r) for r in
                response.GetRows()]
        avg = sum(vals) / len(vals)
        return avg, vals
      else:
        raise ValueError('no rows')
    else:
      raise AuthError('unauthorized')
  else:
    raise ValueError('wrong encoding')
else:
  raise RpcError(rpc.ErrorText())
```

Code with less nesting

```
response = stub.Call(request, rpc)

if rpc.status != pywraprpc.RPC.OK:
  raise RpcError(rpc.ErrorText())

if not response.GetAuthorizedUser():
  raise ValueError('wrong encoding')

if response.GetEnc() != 'utf-8':
  raise AuthError('unauthorized')

if not response.GetRows():
  raise ValueError('no rows')

vals = [ParseRow(r) for r in
        response.GetRows()]
avg = sum(vals) / len(vals)
return avg, vals
```

Answer: the "*wrong encoding*" and "*unauthorized*" errors are swapped. **This bug is easier to see in the refactored version, since the checks occur right as the errors are handled.**

The refactoring technique shown above is known as *guard clauses*. A guard clause checks a criterion and fails fast if it is not met. It decouples the computational logic from the error logic. By removing the cognitive gap between error checking and handling, it frees up mental processing power. As a result, **the refactored version is much easier to read and maintain.**

Here are some rules of thumb for reducing nesting in your code:

- Keep conditional blocks short. It increases readability by keeping things local.
- Consider refactoring when your loops and branches are more than 2 levels deep.
- Think about moving nested logic into separate functions. For example, if you need to loop through a list of objects, each of which contains a list (such as a protocol buffer with repeated fields), you can define a function to process each object instead of using a double-nested loop.

Reducing nesting results in more readable code, which leads to discoverable bugs, faster developer iteration, and increased stability. When you can, simplify!

Flattening Arrow Code
Jeff Atwood of the *Coding Horror* blog shares his experiences about reducing nesting.
go/coding-horror-arrow-code

Join the code-health mailing list
Share your knowledge and learn from Code Health enthusiasts from across the company.
g/code-health

Learn more about Code Health: go/CodeHealth

Read all TotTs online: go/tott

Example TotT:
Don't Put Logic in Tests

Test that uses logic:

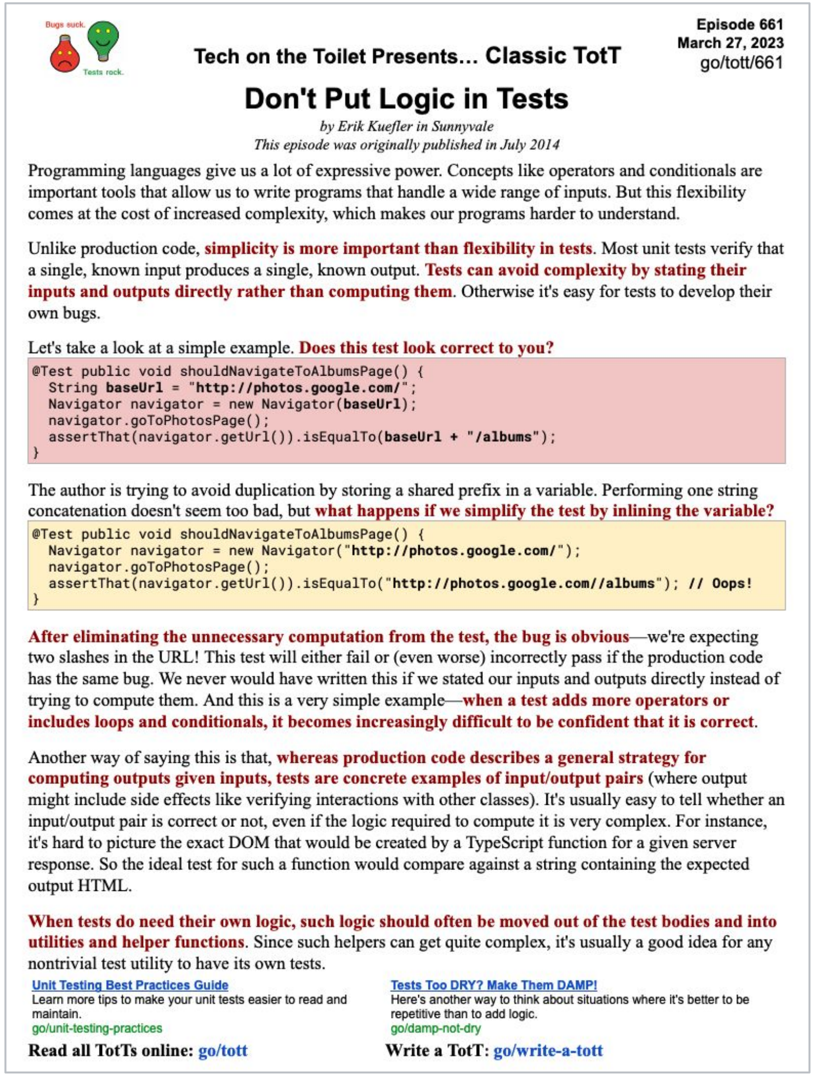
```
String baseUrl = "http://photos.google.com/";

assertThat(navigator.getUrl())
    .isEqualTo(baseUrl + "/albums")
```

Test that doesn't use logic:

```
assertThat(navigator.getUrl())
    .isEqualTo("http://photos.google.com//albums")
```

[Read it online]



Bugs suck. Tests rock.

Episode 661
March 27, 2023
go/tott/661

Tech on the Toilet Presents... Classic TotT

Don't Put Logic in Tests

by Erik Kuefler in Sunnyvale
This episode was originally published in July 2014

Programming languages give us a lot of expressive power. Concepts like operators and conditionals are important tools that allow us to write programs that handle a wide range of inputs. But this flexibility comes at the cost of increased complexity, which makes our programs harder to understand.

Unlike production code, **simplicity is more important than flexibility in tests**. Most unit tests verify that a single, known input produces a single, known output. **Tests can avoid complexity by stating their inputs and outputs directly rather than computing them**. Otherwise it's easy for tests to develop their own bugs.

Let's take a look at a simple example. **Does this test look correct to you?**

```
@Test public void shouldNavigateToAlbumsPage() {
  String baseUrl = "http://photos.google.com/";
  Navigator navigator = new Navigator(baseUrl);
  navigator.goToPhotosPage();
  assertThat(navigator.getUrl()).isEqualTo(baseUrl + "/albums");
}
```

The author is trying to avoid duplication by storing a shared prefix in a variable. Performing one string concatenation doesn't seem too bad, but **what happens if we simplify the test by inlining the variable?**

```
@Test public void shouldNavigateToAlbumsPage() {
  Navigator navigator = new Navigator("http://photos.google.com/");
  navigator.goToPhotosPage();
  assertThat(navigator.getUrl()).isEqualTo("http://photos.google.com//albums"); // Oops!
}
```

After eliminating the unnecessary computation from the test, the bug is obvious—we're expecting two slashes in the URL! This test will either fail or (even worse) incorrectly pass if the production code has the same bug. We never would have written this if we stated our inputs and outputs directly instead of trying to compute them. And this is a very simple example—**when a test adds more operators or includes loops and conditionals, it becomes increasingly difficult to be confident that it is correct.**

Another way of saying this is that, **whereas production code describes a general strategy for computing outputs given inputs, tests are concrete examples of input/output pairs** (where output might include side effects like verifying interactions with other classes). It's usually easy to tell whether an input/output pair is correct or not, even if the logic required to compute it is very complex. For instance, it's hard to picture the exact DOM that would be created by a TypeScript function for a given server response. So the ideal test for such a function would compare against a string containing the expected output HTML.

When tests do need their own logic, such logic should often be moved out of the test bodies and into utilities and helper functions. Since such helpers can get quite complex, it's usually a good idea for any nontrivial test utility to have its own tests.

Unit Testing Best Practices Guide
Learn more tips to make your unit tests easier to read and maintain.
go/unit-testing-practices

Tests Too DRY? Make Them DAMP!
Here's another way to think about situations where it's better to be repetitive than to add logic.
go/damp-not-dry

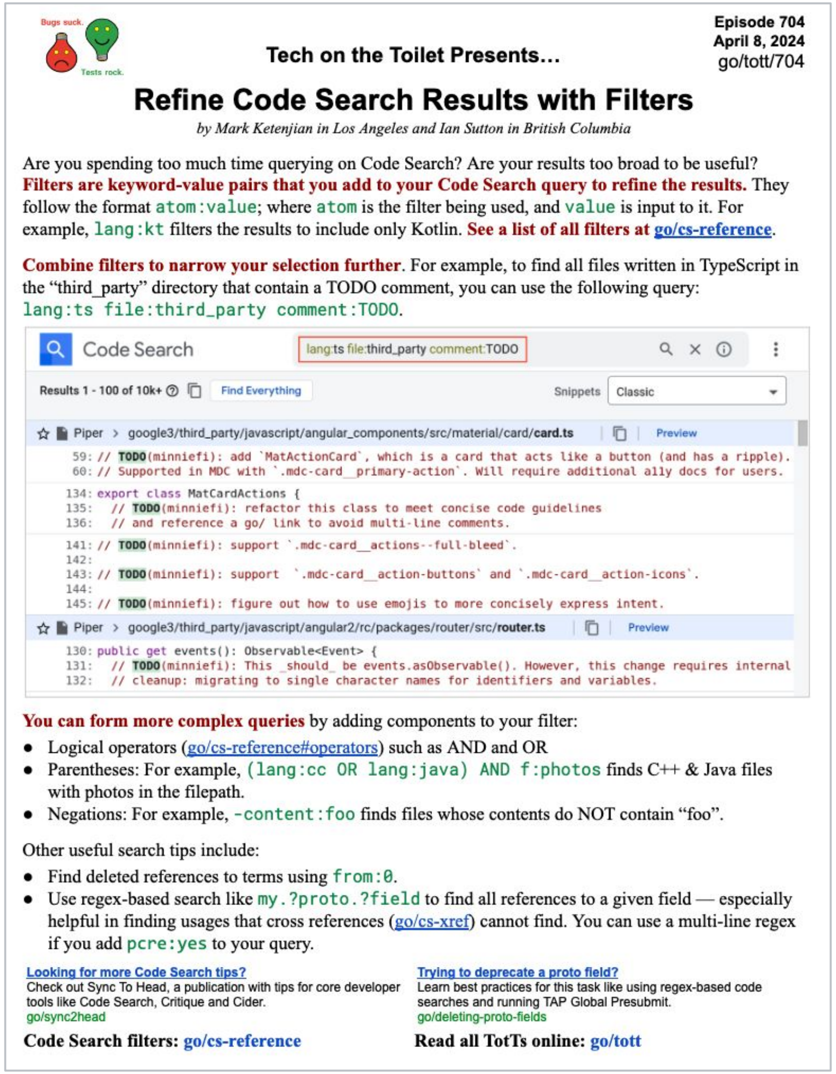
Read all TotTs online: go/tott **Write a TotT:** go/write-a-tott

Example TotT: *Refine Code Search Results with Filters*

Tips for filtering search results in Code Search, an internal web app at Google.

Example filter:

`(lang:cc OR lang:java) AND file:photos`



Bugs suck. Tests rock.

Tech on the Toilet Presents...

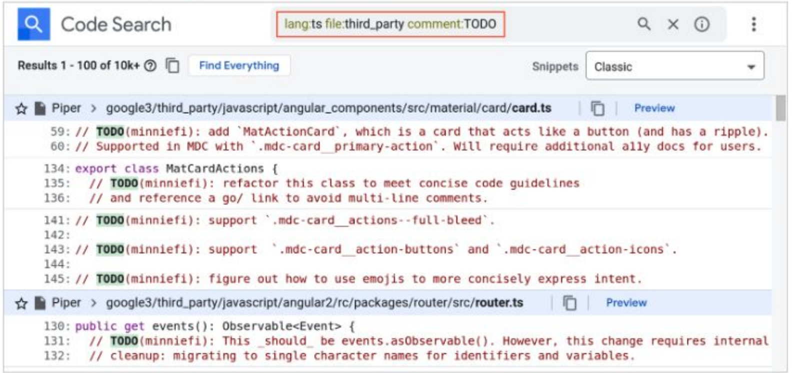
Episode 704
April 8, 2024
go/tott/704

Refine Code Search Results with Filters

by Mark Ketenjian in Los Angeles and Ian Sutton in British Columbia

Are you spending too much time querying on Code Search? Are your results too broad to be useful? **Filters are keyword-value pairs that you add to your Code Search query to refine the results.** They follow the format `atom:value`; where `atom` is the filter being used, and `value` is input to it. For example, `lang:kt` filters the results to include only Kotlin. **See a list of all filters at go/cs-reference.**

Combine filters to narrow your selection further. For example, to find all files written in TypeScript in the "third_party" directory that contain a TODO comment, you can use the following query: `lang:ts file:third_party comment:TODO`.



Code Search `lang:ts file:third_party comment:TODO`

Results 1 - 100 of 10k+ Find Everything Snippets Classic

Piper > google3/third_party/javascript/angular_components/src/material/card/**card.ts** Preview

```
// TODO(minniefi): add `MatActionCard`, which is a card that acts like a button (and has a ripple).
// Supported in MDC with `.mdc-card__primary-action`. Will require additional ally docs for users.

export class MatCardActions {
  // TODO(minniefi): refactor this class to meet concise code guidelines
  // and reference a go/ link to avoid multi-line comments.

// TODO(minniefi): support `.mdc-card__actions--full-bleed`.

// TODO(minniefi): support  `.mdc-card__action-buttons` and `.mdc-card__action-icons`.

// TODO(minniefi): figure out how to use emojis to more concisely express intent.
```

Piper > google3/third_party/javascript/angular2/rc/packages/router/src/**router.ts** Preview

```
public get events(): Observable<Event> {
  // TODO(minniefi): This _should_ be events.asObservable(). However, this change requires internal
  // cleanup: migrating to single character names for identifiers and variables.
```

You can form more complex queries by adding components to your filter:

- Logical operators (go/cs-reference#operators) such as AND and OR
- Parentheses: For example, `(lang:cc OR lang:java) AND f:photos` finds C++ & Java files with photos in the filepath.
- Negations: For example, `-content:foo` finds files whose contents do NOT contain "foo".

Other useful search tips include:

- Find deleted references to terms using `from:0`.
- Use regex-based search like `my.?proto.?field` to find all references to a given field — especially helpful in finding usages that cross references (go/cs-xref) cannot find. You can use a multi-line regex if you add `pcre:yes` to your query.

Looking for more Code Search tips?
Check out Sync To Head, a publication with tips for core developer tools like Code Search, Critique and Cider.
go/sync2head

Trying to deprecate a proto field?
Learn best practices for this task like using regex-based code searches and running TAP Global Presubmit.
go/deleting-proto-fields

Code Search filters: go/cs-reference

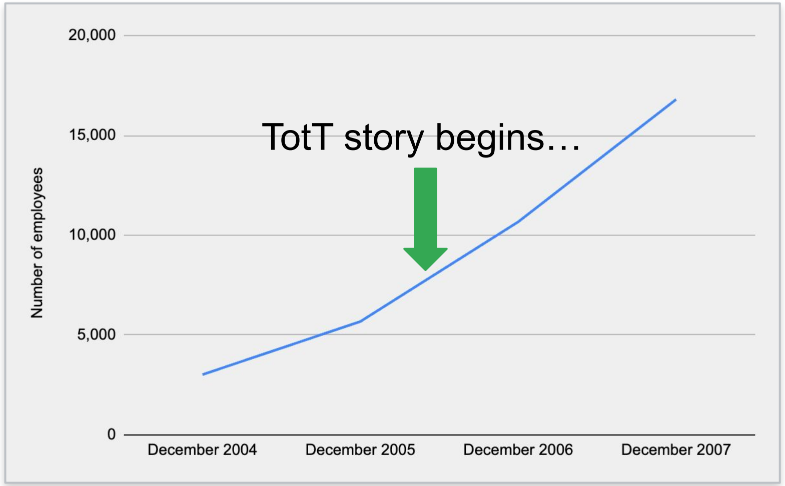
Read all TotTs online: go/tott

Why TotT Was Created

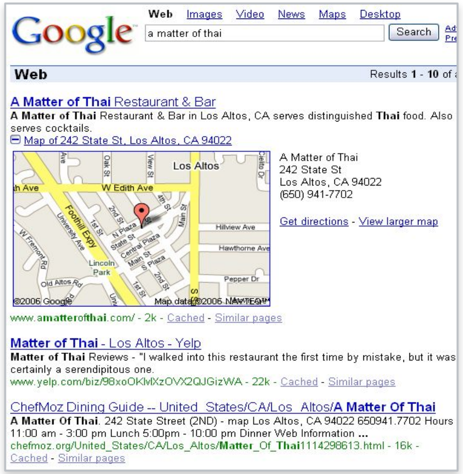
A bottom-up approach to
make a culture change

TotT Origin Story

2006: Google was growing rapidly



A group of Google engineers (“Testing Grouplet”) wanted to spread knowledge about unit testing, so they brainstormed ideas


Web Images Video News Maps Desktop

a matter of thai

Search

Web Results **1** - **10** of

A Matter of Thai Restaurant & Bar
A Matter of Thai Restaurant & Bar in Los Altos, CA serves distinguished **Thai** food. Also serves cocktails.
Map of 242 State St, Los Altos, CA 94022

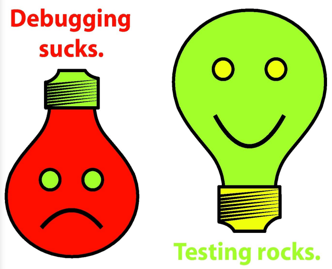
A Matter of Thai
242 State St
Los Altos, CA 94022
(650) 941-7702

Get directions - View larger map

www.**amatterofthai**.com/ - 2k - Cached - Similar pages

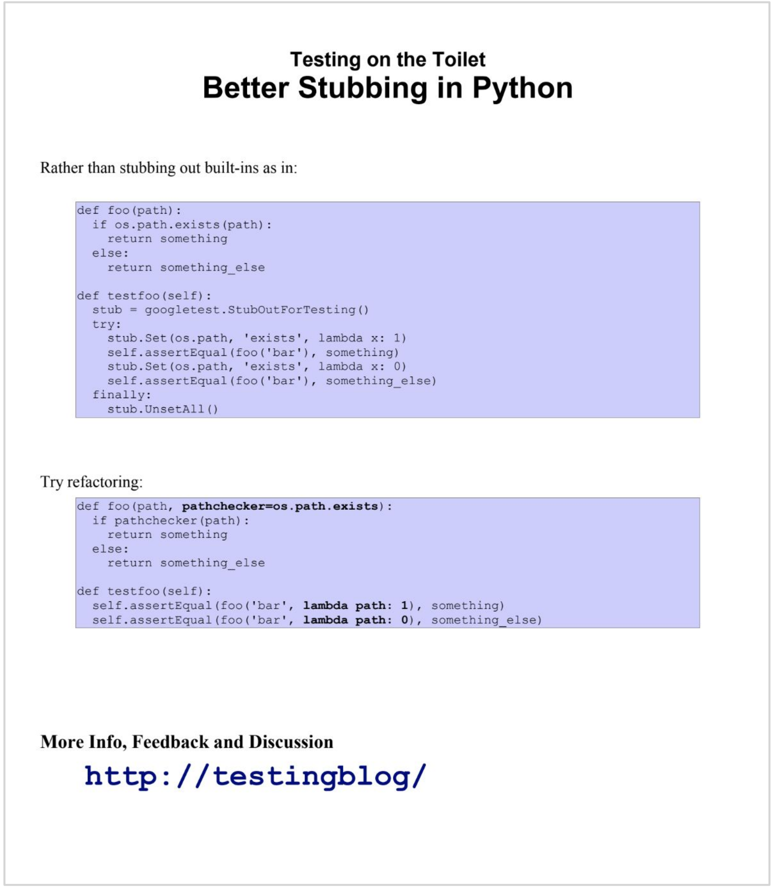
Matter of Thai - Los Altos - Yelp
Matter of Thai Reviews - "I walked into this restaurant the first time by mistake, but it was certainly a serendipitous one.
www.yelp.com/biz/98xoOKlvlXzOVX2QJGizWA - 22k - Cached - Similar pages

ChefMoz Dining Guide -- United States/CA/Los Altos/**A Matter Of Thai**
A Matter Of Thai. 242 State Street (2ND) - map Los Altos, CA 94022 650941.7702 Hours 11:00 am - 3:00 pm Lunch 5:00pm - 10:00 pm Dinner Web Information **...**
chefmoz.org/United_States/CA/Los_Altos/**Matter_Of_Thai**1114298613.html - 16k - Cached - Similar pages



TotT Origin Story

The Testing Grouplet came up with the idea of posting tips about unit testing in bathrooms:
"Testing on the Toilet"



Testing on the Toilet
Better Stubbing in Python

Rather than stubbing out built-ins as in:

```
def foo(path):
  if os.path.exists(path):
    return something
  else:
    return something_else

def testfoo(self):
  stub = googletest.StubOutForTesting()
  try:
    stub.Set(os.path, 'exists', lambda x: 1)
    self.assertEqual(foo('bar'), something)
    stub.Set(os.path, 'exists', lambda x: 0)
    self.assertEqual(foo('bar'), something_else)
  finally:
    stub.UnsetAll()
```

Try refactoring:

```
def foo(path, pathchecker=os.path.exists):
  if pathchecker(path):
    return something
  else:
    return something_else

def testfoo(self):
  self.assertEqual(foo('bar', lambda path: 1), something)
  self.assertEqual(foo('bar', lambda path: 0), something_else)
```

More Info, Feedback and Discussion

http://testingblog/

TotT Origin Story

- Engineers across Google volunteered to start taping it to bathrooms stalls
- Reached most of the several thousand engineers at Google

[Learn more about the creation of TotT](#)

Courtesy: Mike Bland (mike-bland.com)

Seeing TotT for the first time!
(A gen AI depiction)

“

I haven't done any coding in a while, but I know a lot about testing. Do you know why?

(moments of silence...)

What, don't you guys read the flyers they put up in the restrooms?

”

Sergey Brin (Google co-founder)
Google all-hands meeting, October 2006

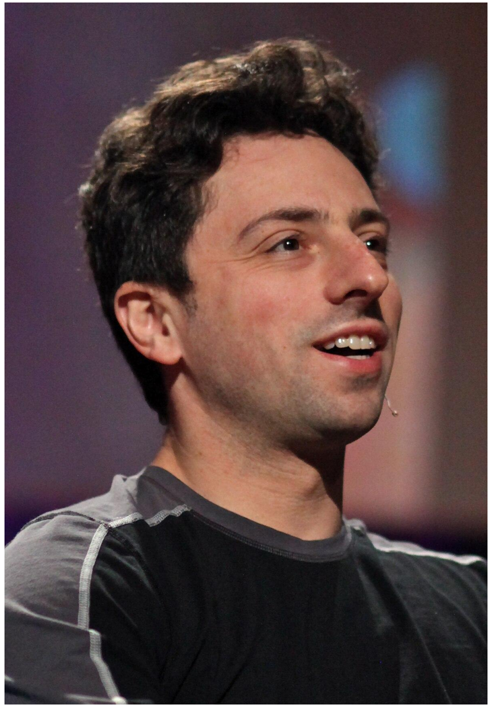


Image source: https://en.wikipedia.org/wiki/Sergey_Brin

TotT Today: By the Numbers

50K+

Audience size (average, per episode)

700+

Episodes published

300K+

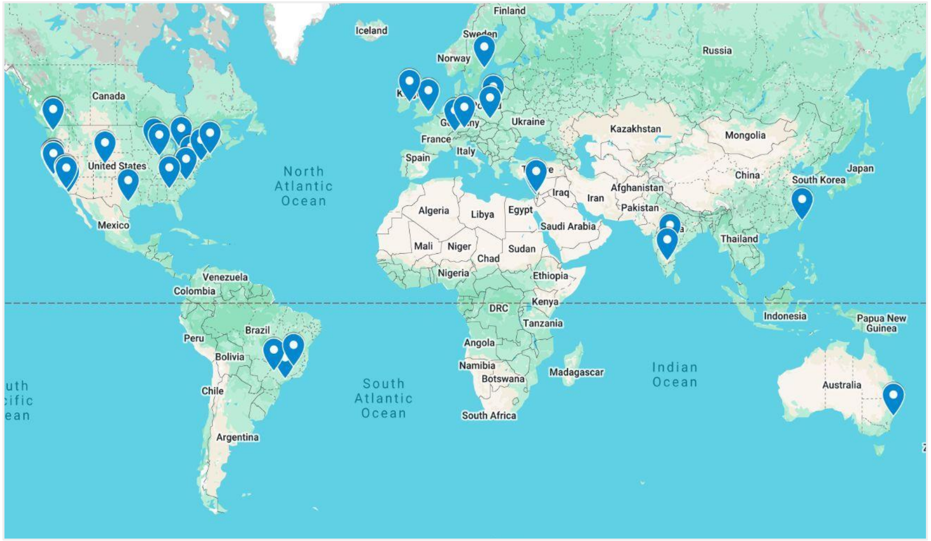
Pageviews for internal TotT website in the past year

550+

Authors

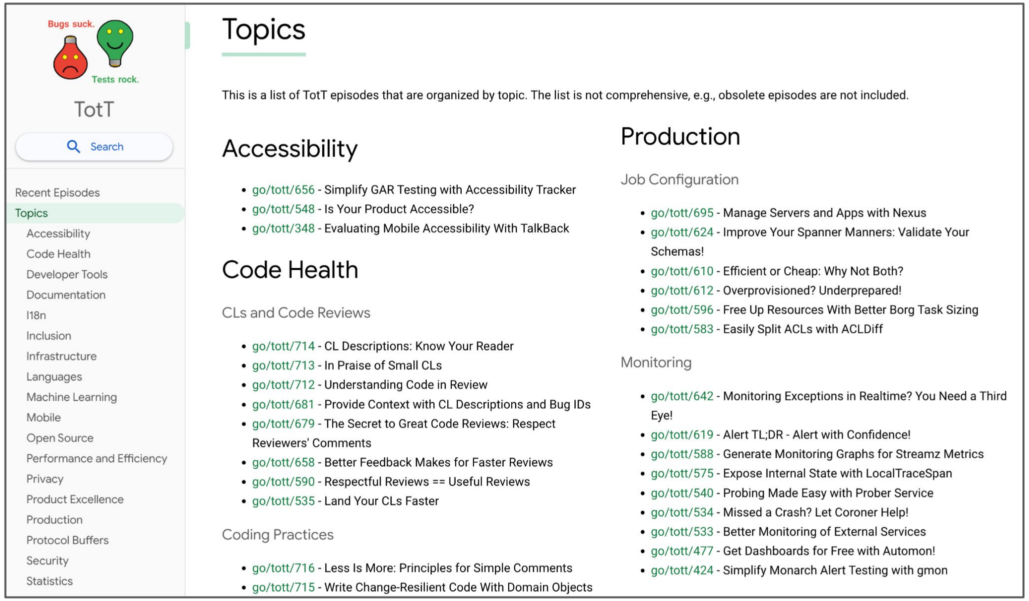
TotT Today: Distribution

- Distributed globally
  - 40+ Google locations
  - 1000s of bathroom stalls
- Posted by the Google Facilities team
- Bi-weekly posting



TotT Today: Audience

- Internal website: Episodes organized by topic
- Internal mailing list: ~10K subscribers
- testing.googleblog.com: 100+ blog posts



Topics

This is a list of TotT episodes that are organized by topic. The list is not comprehensive, e.g., obsolete episodes are not included.

Accessibility

- go/tott/656 - Simplify GAR Testing with Accessibility Tracker
- go/tott/548 - Is Your Product Accessible?
- go/tott/348 - Evaluating Mobile Accessibility With TalkBack

Code Health

CLs and Code Reviews

- go/tott/714 - CL Descriptions: Know Your Reader
- go/tott/713 - In Praise of Small CLs
- go/tott/712 - Understanding Code in Review
- go/tott/681 - Provide Context with CL Descriptions and Bug IDs
- go/tott/679 - The Secret to Great Code Reviews: Respect Reviewers' Comments
- go/tott/658 - Better Feedback Makes for Faster Reviews
- go/tott/590 - Respectful Reviews == Useful Reviews
- go/tott/535 - Land Your CLs Faster

Coding Practices

- go/tott/716 - Less Is More: Principles for Simple Comments
- go/tott/715 - Write Change-Resilient Code With Domain Objects

Production

Job Configuration

- go/tott/695 - Manage Servers and Apps with Nexus
- go/tott/624 - Improve Your Spanner Manners: Validate Your Schemas!
- go/tott/610 - Efficient or Cheap: Why Not Both?
- go/tott/612 - Overprovisioned? Underprepared!
- go/tott/596 - Free Up Resources With Better Borg Task Sizing
- go/tott/583 - Easily Split ACLs with ACLDiff

Monitoring

- go/tott/642 - Monitoring Exceptions in Realtime? You Need a Third Eye!
- go/tott/619 - Alert TL;DR - Alert with Confidence!
- go/tott/588 - Generate Monitoring Graphs for Streamz Metrics
- go/tott/575 - Expose Internal State with LocalTraceSpan
- go/tott/540 - Probing Made Easy with Prober Service
- go/tott/534 - Missed a Crash? Let Coroner Help!
- go/tott/533 - Better Monitoring of External Services
- go/tott/477 - Get Dashboards for Free with Automon!
- go/tott/424 - Simplify Monarch Alert Testing with gmon

TotT Impact

An **efficient** and **effective** way to spread software development knowledge at Google

TotT Impact: Authoritative Source For Best Practices

Many episodes are cited hundreds of times, such as in **code reviews**

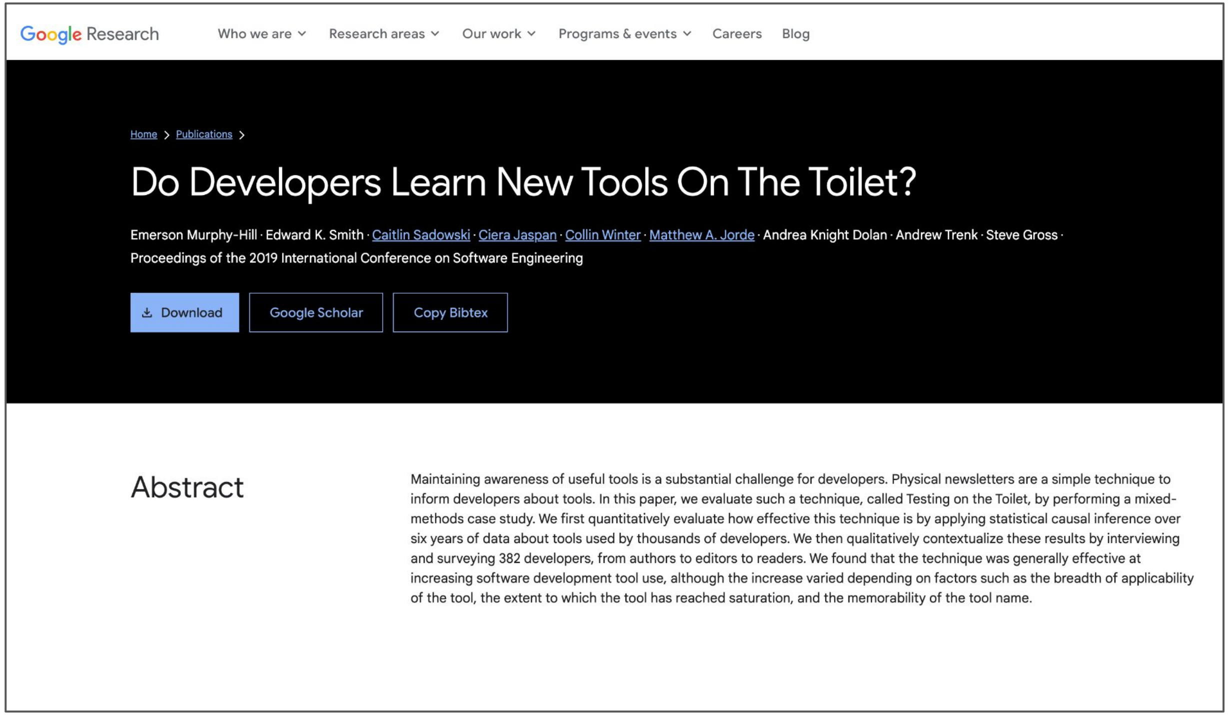
"I agree with this sentiment from go/tott/465: Sometimes the need for a comment can be a sign that the code should be refactored."
To Comment or Not to Comment?

"Can we move setup into the tests themselves so that we keep our tests DAMP? … go/tott/598"
Tests Too DRY? Make Them DAMP!

"Prefer to set these directly in the test as they help explain behaviors. go/tott/677"
Keep Cause and Effect Clear

TotT Impact: Increased Awareness of Development Tools

TotT research paper (2019): *“We found that the technique was generally effective at increasing software development tool use.”*



Do Developers Learn New Tools On The Toilet?

Emerson Murphy-Hill · Edward K. Smith · Caitlin Sadowski · Ciera Jaspan · Collin Winter · Matthew A. Jorde · Andrea Knight Dolan · Andrew Trenk · Steve Gross ·
Proceedings of the 2019 International Conference on Software Engineering

Abstract

Maintaining awareness of useful tools is a substantial challenge for developers. Physical newsletters are a simple technique to inform developers about tools. In this paper, we evaluate such a technique, called Testing on the Toilet, by performing a mixed-methods case study. We first quantitatively evaluate how effective this technique is by applying statistical causal inference over six years of data about tools used by thousands of developers. We then qualitatively contextualize these results by interviewing and surveying 382 developers, from authors to editors to readers. We found that the technique was generally effective at increasing software development tool use, although the increase varied depending on factors such as the breadth of applicability of the tool, the extent to which the tool has reached saturation, and the memorability of the tool name.

TotT Impact: Testimonials

"I love TotT! … it is one of my favorite things about working at Google!"

"TotT definitely factored into my decision to accept the offer from Google."

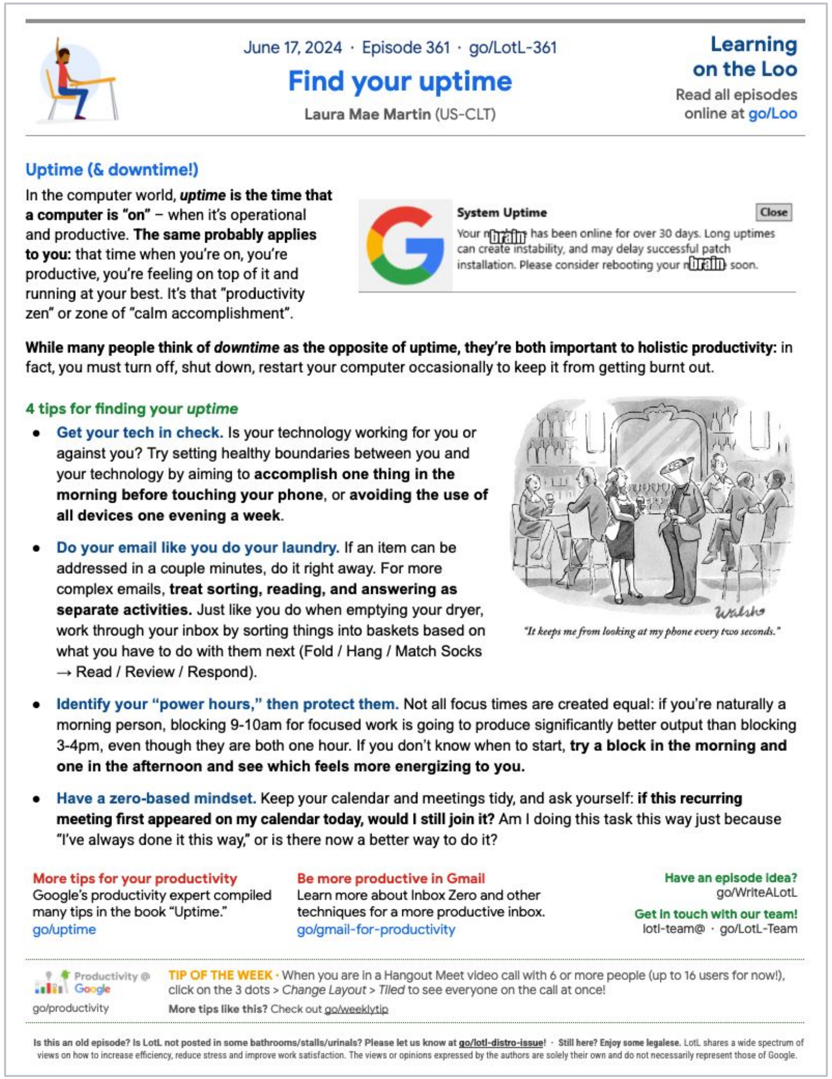
"This helps in my day to day work and helps me grow as engineer."

TotT Today: Inspires Other Publications

Learning on the Loo: Short articles about productivity, career development, and personal wellbeing.

Posted alongside TotT

Read: [The inside story of how Google bathrooms became classrooms]()



June 17, 2024 · Episode 361 · go/LotL-361

Find your uptime

Laura Mae Martin (US-CLT)

Learning on the Loo
Read all episodes online at go/Loo

Uptime (& downtime!)

In the computer world, ***uptime* is the time that a computer is "on"** – when it's operational and productive. **The same probably applies to you:** that time when you're on, you're productive, you're feeling on top of it and running at your best. It's that "productivity zen" or zone of "calm accomplishment".

System Uptime Close
Your brain has been online for over 30 days. Long uptimes can create instability, and may delay successful patch installation. Please consider rebooting your brain soon.

While many people think of *downtime* as the opposite of uptime, they're both important to holistic productivity: in fact, you must turn off, shut down, restart your computer occasionally to keep it from getting burnt out.

4 tips for finding your *uptime*

- **Get your tech in check.** Is your technology working for you or against you? Try setting healthy boundaries between you and your technology by aiming to **accomplish one thing in the morning before touching your phone**, or **avoiding the use of all devices one evening a week**.
- **Do your email like you do your laundry.** If an item can be addressed in a couple minutes, do it right away. For more complex emails, **treat sorting, reading, and answering as separate activities.** Just like you do when emptying your dryer, work through your inbox by sorting things into baskets based on what you have to do with them next (Fold / Hang / Match Socks → Read / Review / Respond).
- **Identify your "power hours," then protect them.** Not all focus times are created equal: if you're naturally a morning person, blocking 9-10am for focused work is going to produce significantly better output than blocking 3-4pm, even though they are both one hour. If you don't know when to start, **try a block in the morning and one in the afternoon and see which feels more energizing to you.**
- **Have a zero-based mindset.** Keep your calendar and meetings tidy, and ask yourself: **if this recurring meeting first appeared on my calendar today, would I still join it?** Am I doing this task this way just because "I've always done it this way," or is there now a better way to do it?

"It keeps me from looking at my phone every two seconds."

More tips for your productivity
Google's productivity expert compiled many tips in the book "Uptime."
go/uptime

Be more productive in Gmail
Learn more about Inbox Zero and other techniques for a more productive inbox.
go/gmail-for-productivity

Have an episode idea?
go/WriteALotL

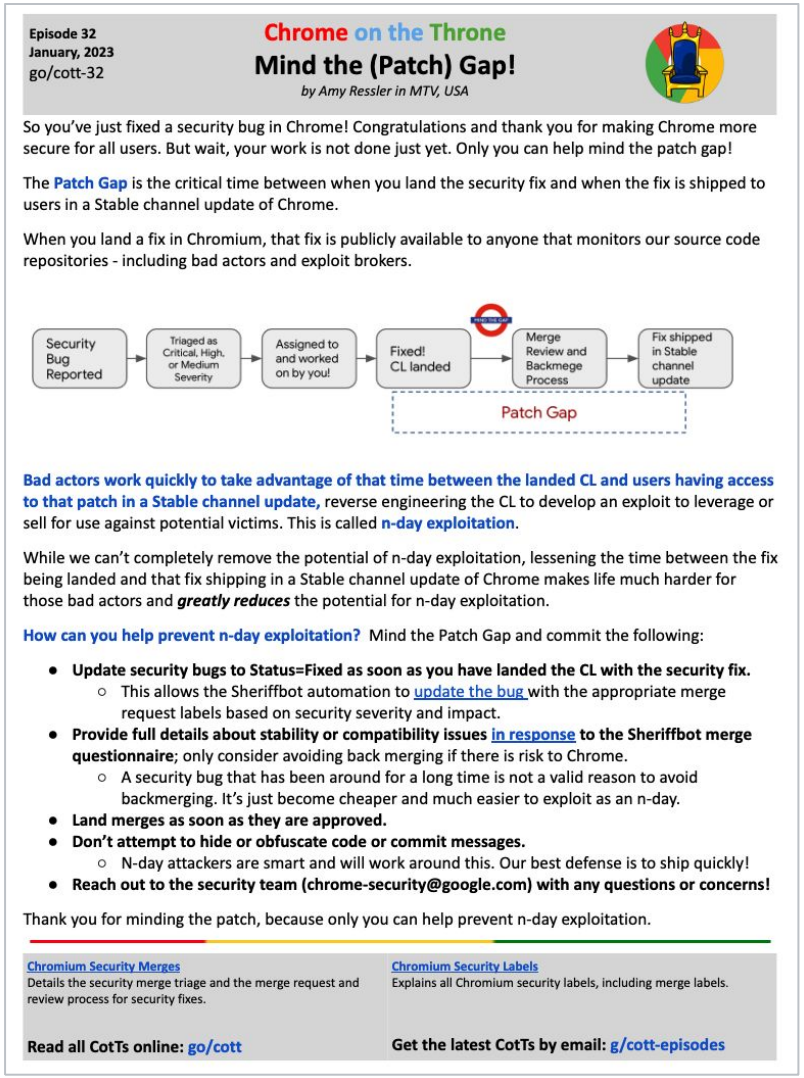
Get in touch with our team!
lotl-team@ · go/LotL-Team

Productivity @ Google
go/productivity

TIP OF THE WEEK · When you are in a Hangout Meet video call with 6 or more people (up to 16 users for now!), click on the 3 dots > *Change Layout* > *Tiled* to see everyone on the call at once!
More tips like this? Check out go/weeklytip

TotT Today: Inspires Other Publications

Chrome on the Throne:
Technical content posted in Chrome team offices



Episode 32
January, 2023
go/cott-32

Chrome on the Throne

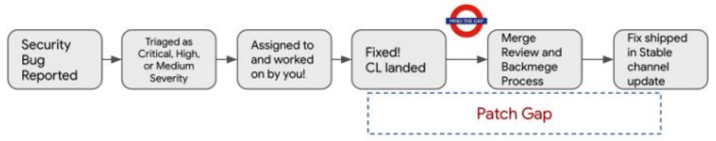
Mind the (Patch) Gap!

by Amy Ressler in MTV, USA

So you've just fixed a security bug in Chrome! Congratulations and thank you for making Chrome more secure for all users. But wait, your work is not done just yet. Only you can help mind the patch gap!

The **Patch Gap** is the critical time between when you land the security fix and when the fix is shipped to users in a Stable channel update of Chrome.

When you land a fix in Chromium, that fix is publicly available to anyone that monitors our source code repositories - including bad actors and exploit brokers.



Security Bug Reported
Triaged as Critical, High, or Medium Severity
Assigned to and worked on by you!
Fixed! CL landed
MIND THE GAP
Merge Review and Backmege Process
Fix shipped in Stable channel update
Patch Gap

Bad actors work quickly to take advantage of that time between the landed CL and users having access to that patch in a Stable channel update, reverse engineering the CL to develop an exploit to leverage or sell for use against potential victims. This is called **n-day exploitation**.

While we can't completely remove the potential of n-day exploitation, lessening the time between the fix being landed and that fix shipping in a Stable channel update of Chrome makes life much harder for those bad actors and ***greatly reduces*** the potential for n-day exploitation.

How can you help prevent n-day exploitation? Mind the Patch Gap and commit the following:

- **Update security bugs to Status=Fixed as soon as you have landed the CL with the security fix.**
  - This allows the Sheriffbot automation to update the bug with the appropriate merge request labels based on security severity and impact.
- **Provide full details about stability or compatibility issues in response to the Sheriffbot merge questionnaire**; only consider avoiding back merging if there is risk to Chrome.
  - A security bug that has been around for a long time is not a valid reason to avoid backmerging. It's just become cheaper and much easier to exploit as an n-day.
- **Land merges as soon as they are approved.**
- **Don't attempt to hide or obfuscate code or commit messages.**
  - N-day attackers are smart and will work around this. Our best defense is to ship quickly!
- **Reach out to the security team (chrome-security@google.com) with any questions or concerns!**

Thank you for minding the patch, because only you can help prevent n-day exploitation.

Chromium Security Merges
Details the security merge triage and the merge request and review process for security fixes.

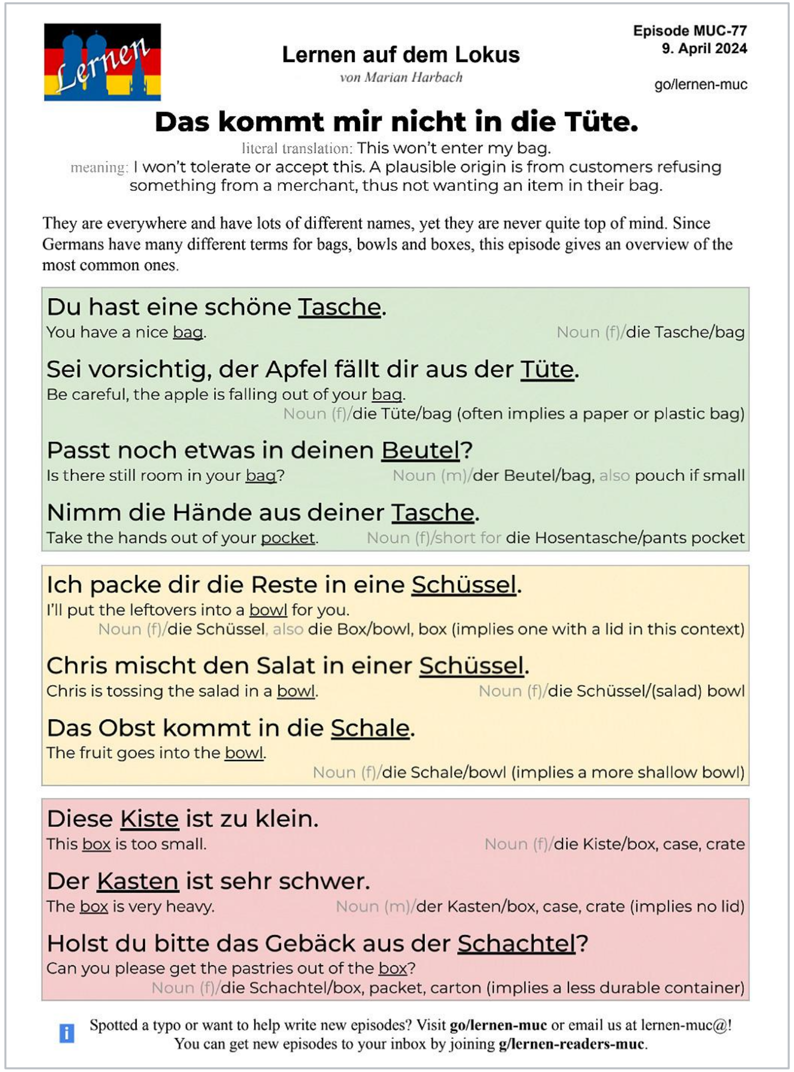
Chromium Security Labels
Explains all Chromium security labels, including merge labels.

Read all CotTs online: go/cott

Get the latest CotTs by email: g/cott-episodes

TotT Today: Inspires Other Publications

Lernen auf dem Lokus: Germany and Switzerland offices only, teaches German



Lernen auf dem Lokus

von Marian Harbach

Episode MUC-77
9. April 2024

go/lernen-muc

Das kommt mir nicht in die Tüte.

literal translation: This won't enter my bag.
meaning: I won't tolerate or accept this. A plausible origin is from customers refusing something from a merchant, thus not wanting an item in their bag.

They are everywhere and have lots of different names, yet they are never quite top of mind. Since Germans have many different terms for bags, bowls and boxes, this episode gives an overview of the most common ones.

Du hast eine schöne <u>Tasche</u>.
You have a nice <u>bag</u>. Noun (f)/die Tasche/bag

Sei vorsichtig, der Apfel fällt dir aus der <u>Tüte</u>.
Be careful, the apple is falling out of your <u>bag</u>.
Noun (f)/die Tüte/bag (often implies a paper or plastic bag)

Passt noch etwas in deinen <u>Beutel</u>?
Is there still room in your <u>bag</u>? Noun (m)/der Beutel/bag, also pouch if small

Nimm die Hände aus deiner <u>Tasche</u>.
Take the hands out of your <u>pocket</u>. Noun (f)/short for die Hosentasche/pants pocket

Ich packe dir die Reste in eine <u>Schüssel</u>.
I'll put the leftovers into a <u>bowl</u> for you.
Noun (f)/die Schüssel, also die Box/bowl, box (implies one with a lid in this context)

Chris mischt den Salat in einer <u>Schüssel</u>.
Chris is tossing the salad in a <u>bowl</u>. Noun (f)/die Schüssel/(salad) bowl

Das Obst kommt in die <u>Schale</u>.
The fruit goes into the <u>bowl</u>.
Noun (f)/die Schale/bowl (implies a more shallow bowl)

Diese <u>Kiste</u> ist zu klein.
This <u>box</u> is too small. Noun (f)/die Kiste/box, case, crate

Der <u>Kasten</u> ist sehr schwer.
The <u>box</u> is very heavy. Noun (m)/der Kasten/box, case, crate (implies no lid)

Holst du bitte das Gebäck aus der <u>Schachtel</u>?
Can you please get the pastries out of the <u>box</u>?
Noun (f)/die Schachtel/box, packet, carton (implies a less durable container)

i Spotted a typo or want to help write new episodes? Visit **go/lernen-muc** or email us at lernen-muc@! You can get new episodes to your inbox by joining **g/lernen-readers-muc**.

Testing on the Toilet
Learning on the Loo
Classroom in the Bathroom
Bidet Bootcamp
Urinal University

TotT Today: Inspires Other Publications

The TotT Team

20% time

Volunteer-driven: ~10 member team dedicates their time to keep the effort running

- Triage proposals
- Review & edit content
- Maintain internal site & external blog posts
- Coordinate with Google facilities

Why TotT is Effective

1. Length
2. Content
3. Audience

The three magic ingredients

Why TotT is Effective: *Length*

- **Limited to one page**
  - Much easier to read, understand, and remember
  - Focuses on the most important points

Why TotT is Effective: *Content*

- **Actionable**
  - Clear takeaways
  - Widely relevant across Google
- **Trusted**
  - Treated as authoritative
  - Rigorous review process

Why TotT is Effective: *Content*

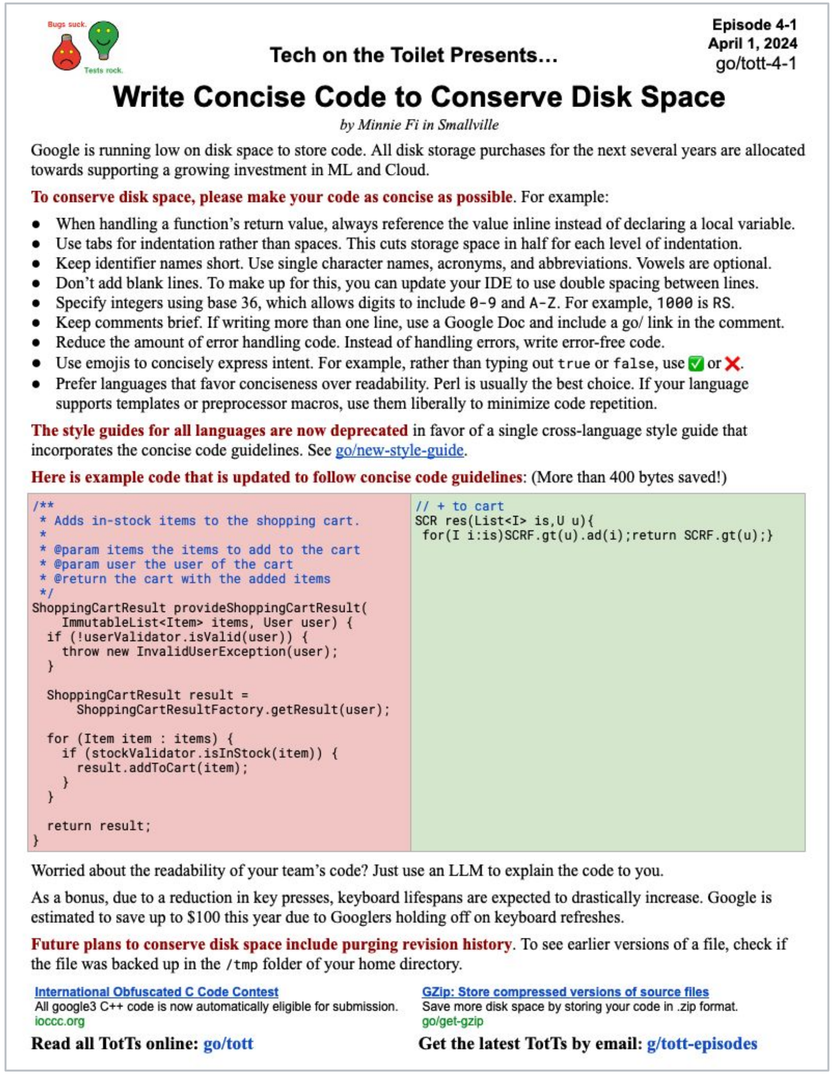
Is TotT trusted too much?
April Fools 2024

“This year's April Fools joke was probably my favorite I've ever seen.”

“OH MY GOD. This made my day.”

“This made me smile and laugh. Thank you!”

All content in this TotT episode is nonsense!



Bugs suck. Tests rock.

Tech on the Toilet Presents…

Episode 4-1
April 1, 2024
go/tott-4-1

Write Concise Code to Conserve Disk Space

by Minnie Fi in Smallville

Google is running low on disk space to store code. All disk storage purchases for the next several years are allocated towards supporting a growing investment in ML and Cloud.

To conserve disk space, please make your code as concise as possible. For example:

- When handling a function’s return value, always reference the value inline instead of declaring a local variable.
- Use tabs for indentation rather than spaces. This cuts storage space in half for each level of indentation.
- Keep identifier names short. Use single character names, acronyms, and abbreviations. Vowels are optional.
- Don’t add blank lines. To make up for this, you can update your IDE to use double spacing between lines.
- Specify integers using base 36, which allows digits to include `0-9` and `A-Z`. For example, `1000` is RS.
- Keep comments brief. If writing more than one line, use a Google Doc and include a go/ link in the comment.
- Reduce the amount of error handling code. Instead of handling errors, write error-free code.
- Use emojis to concisely express intent. For example, rather than typing out `true` or `false`, use ✅ or ❌.
- Prefer languages that favor conciseness over readability. Perl is usually the best choice. If your language supports templates or preprocessor macros, use them liberally to minimize code repetition.

The style guides for all languages are now deprecated in favor of a single cross-language style guide that incorporates the concise code guidelines. See go/new-style-guide.

Here is example code that is updated to follow concise code guidelines: (More than 400 bytes saved!)

```
/**
 * Adds in-stock items to the shopping cart.
 *
 * @param items the items to add to the cart
 * @param user the user of the cart
 * @return the cart with the added items
 */
ShoppingCartResult provideShoppingCartResult(
    ImmutableList<Item> items, User user) {
  if (!userValidator.isValid(user)) {
    throw new InvalidUserException(user);
  }

  ShoppingCartResult result =
      ShoppingCartResultFactory.getResult(user);

  for (Item item : items) {
    if (stockValidator.isInStock(item)) {
      result.addToCart(item);
    }
  }

  return result;
}
```

```
// + to cart
SCR res(List<I> is,U u){
 for(I i:is)SCRF.gt(u).ad(i);return SCRF.gt(u);}
```

Worried about the readability of your team’s code? Just use an LLM to explain the code to you.

As a bonus, due to a reduction in key presses, keyboard lifespans are expected to drastically increase. Google is estimated to save up to $100 this year due to Googlers holding off on keyboard refreshes.

Future plans to conserve disk space include purging revision history. To see earlier versions of a file, check if the file was backed up in the `/tmp` folder of your home directory.

International Obfuscated C Code Contest
All google3 C++ code is now automatically eligible for submission.
ioccc.org

GZip: Store compressed versions of source files
Save more disk space by storing your code in .zip format.
go/get-gzip

Read all TotTs online: go/tott

Get the latest TotTs by email: g/tott-episodes

Why TotT is Effective: *Audience*

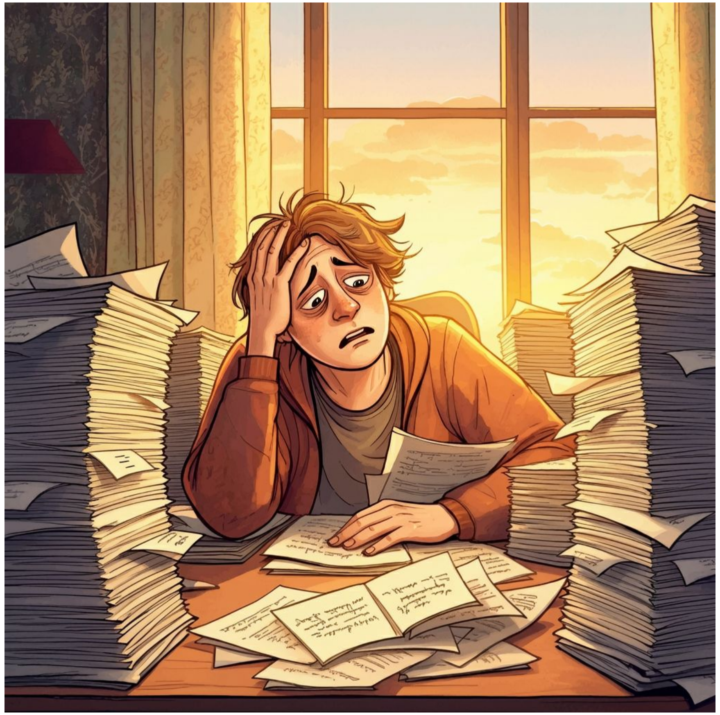
- **Read by most engineers at Google**
  - Captive audience
  - Global distribution

Lessons Learned

1. *Keep It Brief*
2. *Scale Your Knowledge Sharing*
3. *Developers Want Best Practices Documentation*

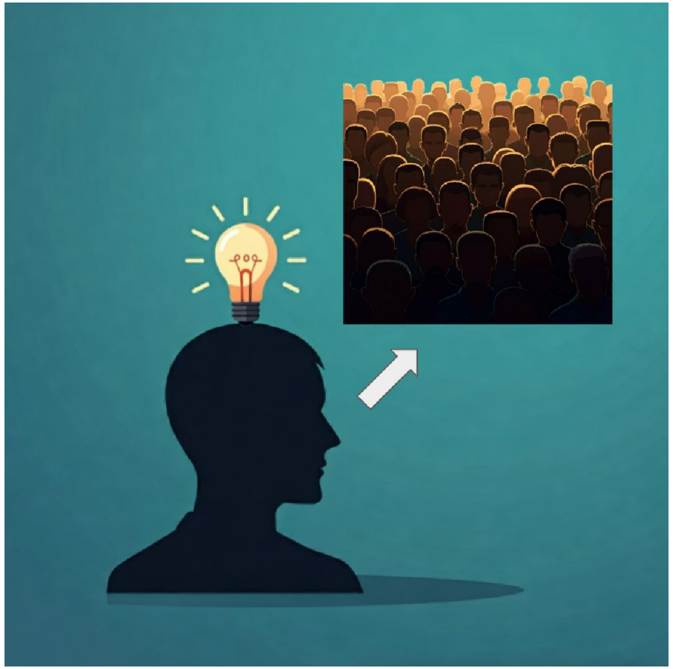
Lesson #1: *Keep It Brief*

- The more content you have, the less likely it is that people will read it
- When sharing information with a large audience, narrow it down to the core concepts
    - What is the minimal amount people need to know?



Lesson #2: *Scale Your Knowledge Sharing*

- Have an opinion about software development? Write it down and share it with others
- Share your knowledge with more people, have more impact
- An email newsletter is an easy way to start. Examples:
  - C++ Tips of the Week
  - Performance Tips of the Week



Lesson #3: *Developers Want Best Practices Documentation*

- It can be hard to find trusted resources about software best practices
- Build a knowledge base
  - Curate a list of links to existing resources
  - Get developers in your company to contribute content
  - Example: [Google code review practices]

Tech on the Toilet (TotT):

Software development topics

Weekly publication

1-pager

For engineers, by engineers

Globally distributed

Effective, efficient

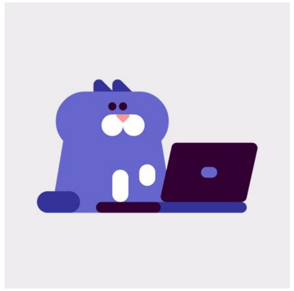
Trusted

Volunteer-run

Thank you

Andrew Trenk
linkedin.com/in/andrewtrenk

Kanu Tewary
linkedin.com/in/kanutewary



Read TotT outside Google:
testing.googleblog.com