

UNCOMMENT YOUR CODE

Miki Tebeka
miki.tebeka@gmail.com

Only 96
slides!

Relax, it will be
over faster than you
think

And there is a game
at the end

I've read an article

(Yes, I read stuff)

Thought it might
interest you

It's about not
commenting your
code

which will make it
better

A quote

“Always write code as if the maintenance programmer were an axe murderer who knows where you live”

- Mike Dunn

He wasn't joking



Meet Donald E.
Knuth

**Don't ask “who is
he?”**

Another quote

“If something can be described in the language itself, one should do that, and not just mention it in a comment”

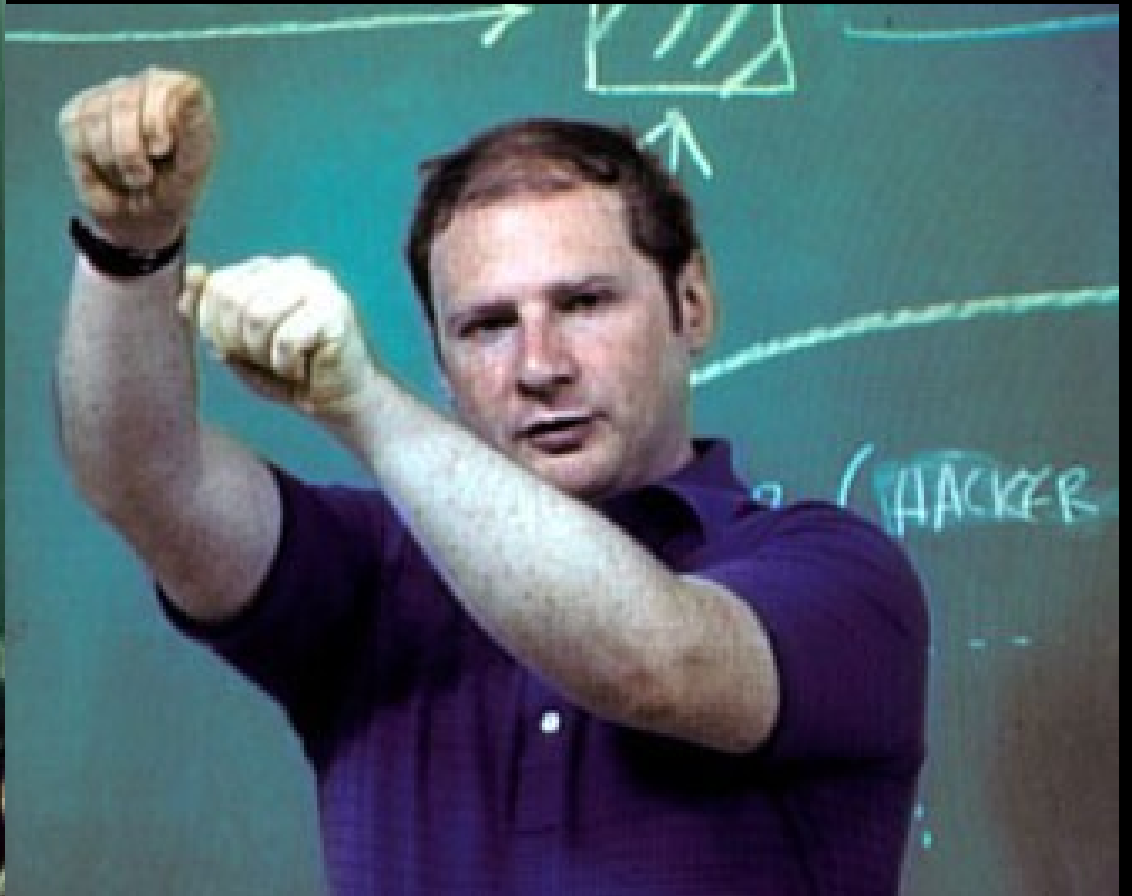


This is Bjarne
Stroustrup

Hope you know
who he is

One more

“Programs must be written for people to read, and only incidentally for machines to execute”



These are Hal
Abelson and Jerry
Sussman

(That's MIT for
you)

Last quote
(for now)

“Do not comment
bad code, rewrite it”



This is Brian W.
Kerningham

Not so frightening
as the others

Let's get dirty

```
Class Badge:
    ''' stores information
        Printed on person
        Bade
    '''
    ...
```


Looks OK?

How about ...

```
class InfoOnBadge:  
    ...
```

Wait!
there's more

```
class InfoOnBadge:  
    def __init__(self, pid, name):  
        self.pid = pid # Person ID  
        self.name = name # Person name
```

can become

```
class InfoOnBadge:
    def __init__(self, personID,
                  personName):
        self.personID = personID
        self.personName = personName
```

We can do the same
for parameters


```
def new_bade_info(id):  
    '''Create new badge information.  
        `id` is the person id  
    '''  
    ....
```

will become

```
def new_badge_info(person_id):  
    ...
```

also

```
def new_badge_info(person_id):  
    # Load from database  
    db = connect_to_database()  
    cursor = db.cursor()  
    ...
```

Can be improved to

```
def new_badge_info(person_id):  
    items = load_from_database(person_id)  
    ...
```

Also delete useless
comments


```
# Open the file  
infile = open( "some_file" )
```

turns to

```
infile = open( "some_file" )
```

Why delete
comments?

We take shortcuts

and write
comments

(which is easy)

to explain obscure
code

instead of writing
clear code

(which is hard)

Problem is

code **always** gets
updated

but comments only
sometimes

So we get obscure
code

with outdated
comments

Not the place we
want to be

However don't take
this to extreme

```
def  
findStockItemsFromOverseasWithInventoryLessThan10():  
    ....
```

we can either

```
def findStockItems(  
    isFromOverseas,  
    maxInventory):  
    ...
```

or state the reason
for the constraints

```
def  
findStockItemsToReplenish( ) :  
    . . .
```



Merriam-Webster
OnLine

- Home
- Visit Our Sites
- Premium Services
- Downloads
- Word of the Day
- Word Games
- Open Dictionary
- Spelling Bee Hive
- Word for the Wise
- Online Store
- Help
- About Us



Also Visit [Unabridged Dictionary](#) [Learner's Dictionary](#) [Word Central for Kids](#) [Collegiate Dictionary](#) [Encyclopædia Britannica](#)

Merriam-Webster
OnLine Search

[Dictionary](#) [Thesaurus](#) [Spanish/English](#)

Search

Merriam-Webster's Online Dictionary

replenish

One entry found for **replenish**.

Main Entry: **re·plen·ish**

Pronunciation: ri-'ple-nish

Function: *verb*

Etymology: Middle English *replenisshen*, from Anglo-French *repleniss-*, stem of *replenir* to fill, from *re-* + *plein* full, from Latin *plenus* -- more at [FULL](#)

transitive verb

1 a : to fill with persons or animals : [STOCK](#) **b** *archaic* : to supply fully : [PERFECT](#) **c** : to fill with inspiration or power : [NOURISH](#)

2 a : to fill or build up again <*replenished* his glass> **b** : to make good : [REPLACE](#)

intransitive verb : to become full : fill up again

- **re·plen·ish·able** /-ni-sh&-b&l/ *adjective*

- **re·plen·ish·er** *noun*

- **re·plen·ish·ment** /-nish-m&nt/ *noun*

Visit [Britannica.com](#) for [more information on "replenish"](#)

Get the [Top 10 Search Results for "replenish"](#)

Comments are not
all bad

Good for the “big
picture”

and

Sometime we do
write tricky code

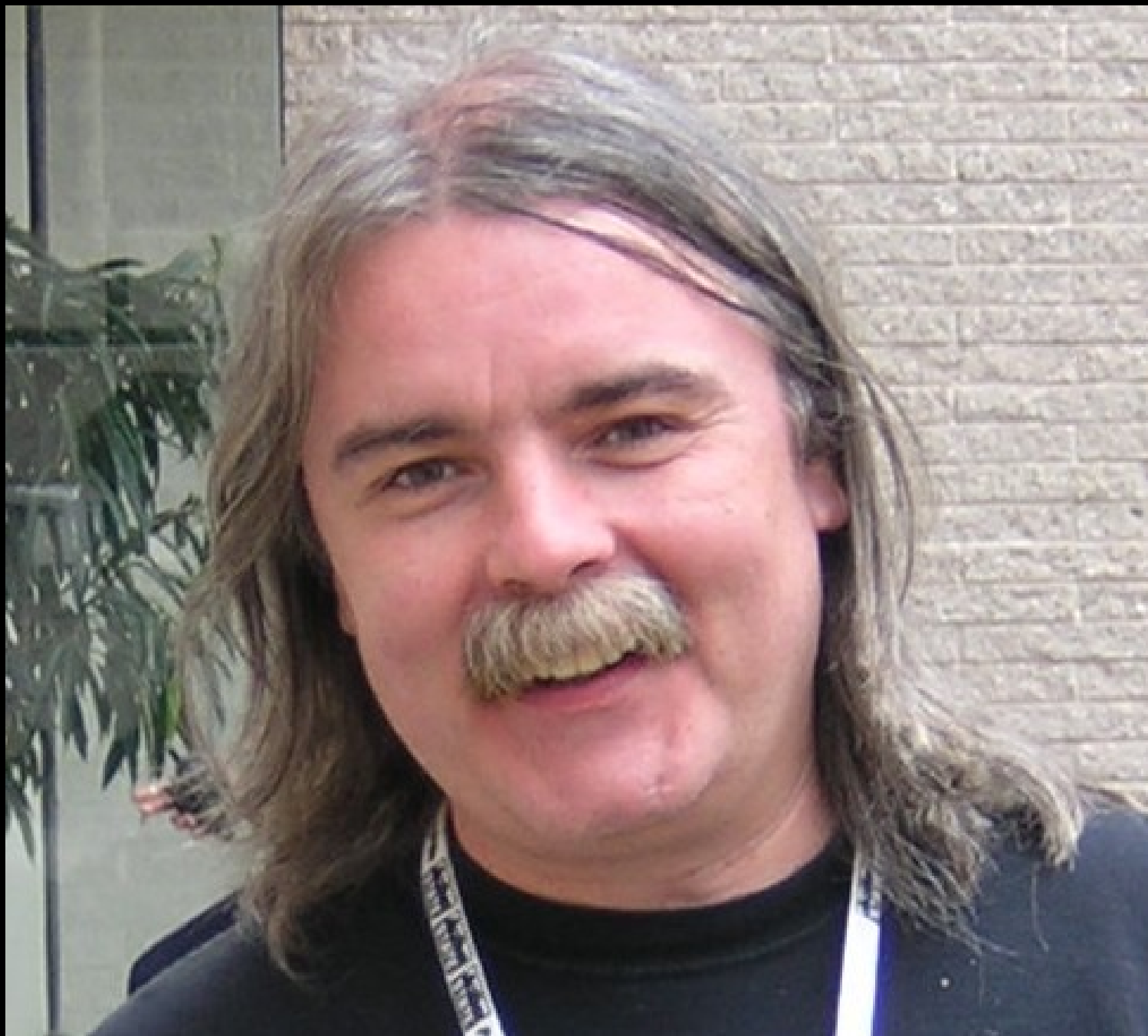
Just make sure

it's absolutely
necessary

To quote a wise
person

“If the
implementation is
hard to explain, it's
a bad idea”

“If the
implementation is
easy to explain, it
may be a good idea”



This is Tim Peters

(Google him)

The quote is from
the Python Zen

Now go and
terminate these
comments

Or



will come after you

with an axe

And he does know
where you live

References

C2 Wiki

Treat Comments with suspicion

“Turning comments
into code”

By Kent Tong

(“Objective View”, 8'th edition)

Python Zen
(here or
“import this”)

“Programming
language inventor
or serial killer”
game

Thank You