Error handling

version: 1.0.0

I was thinking, if the error exists between keyboard and chair, I want the strictest failure mode to both catch it and force me to do things right the first time.

Andai @ https://news.ycombinator.com/item?id=22833601

Agenda

What’s worse than crashing?

Error handling in your code

Tool and compiler help

Error tolerant architecture

War stories

What’s worse than crashing?

Crash scenarios, in order from best to worst:

1. Application works as expected and never crashes.
2. Application crashes due to rare bugs that nobody notices or cares about.
3. Application crashes due to a commonly encountered bug.
4. Application deadlocks and stops responding due to a common bug.
5. Application crashes long after the original bug.
6. **Application causes data loss and/or corruption.**

There's a natural tension between...

- failing *immediately* when your program encounters a problem, eg "fail fast"
- attempting to recover from the failure state and proceed normally

Source: https://blog.codinghorror.com/whats-worse-than-crashing/

Error handling in your code

Checked exceptions

```
InputStream in = null;
try {
  in = new FileInputStream("simple.csv");
  BufferedReader buf = new BufferedReader(new InputStreamReader(in));
} finally {
  if (in != null) {
    try {
      in.close();
    } catch (IOException e) {}
  }
}
```

Compiler helps you handle exceptions when calling methods

Exceptions

- Only catch exceptions you know you can handle
- Don’t
  - `try {`
    `…`
    `} catch (Exception) { }`
    Unless you log and rethrow
  - use exceptions to handle logic errors `NullReferenceException`
  - Don’t use exceptions to emulate control-flow

Get Compiler to help with error handling

- Out parameters in C#

```
public static bool TryParse (string s, out int result);
```

- Create types that captures errors

```
public abstract class Option<T> {
    public abstract bool Exists();
    public abstract T Get();
}
```

```
public class Some<T> : Option<T> {
    private T t;
    public Some(T t) {
        this.t = t;
    }
    public override bool Exists() => true;
    public override T Get() => t;
}

public class None<T> : Option<T> {
    public override bool Exists() => false;
    public override T Get() => throw new Exception();
}
```

- Could also be Either<Error, Success>

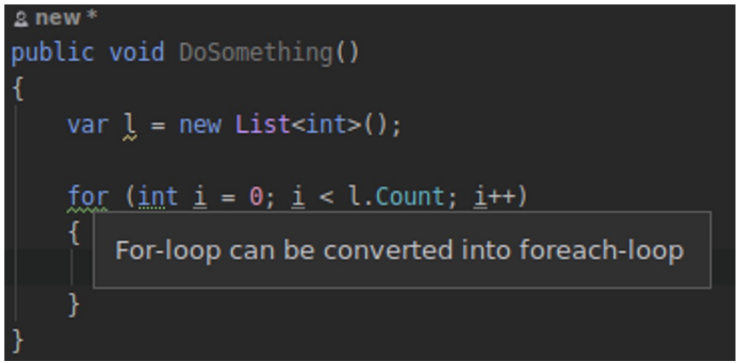
Dynamic type languages

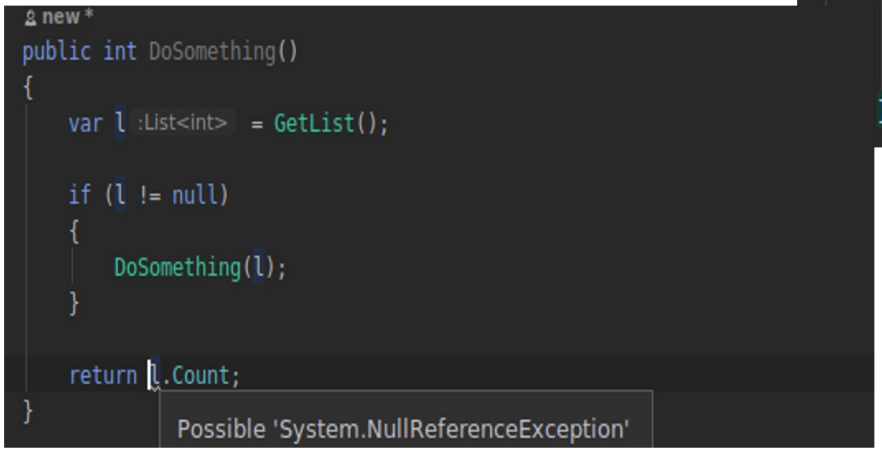
- Examples JavaScript, PHP, Python, …
- Variable can be assigned and re-assigned to anything
- No check for return type of function matches or are equal
- Often just try handling values in some way

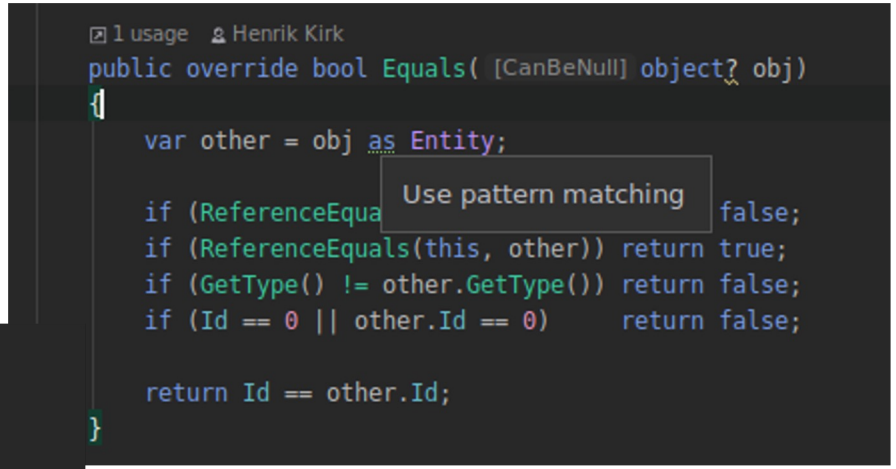
	true	false	1	0	-1	"1"	"0"	"-1"	"true"	"false"	null	"foobar"	"4779"	"0x12AB"	""
true		false	true	false	true	true	false	true	true	true	false	true	true	true	false
false	false		false	true	false	false	true	false	false	false	true	false	false	false	true
1	true	false		false	false	true	false	false	false	false	false	false	false	false	false
0	false	true	false		false	false	true	false	true	true	true	true	false	true	true
-1	true	false	false	false		false	false	true	false	false	false	false	false	false	false
"1"	true	false	true	false	false		false	false	false	false	false	false	false	false	false
"0"	false	true	false	true	false	false		false	false	false	false	false	false	false	false
"-1"	true	false	false	false	true	false	false		false	false	false	false	false	false	false
"true"	true	false	false	true	false	false	false	false		false	false	false	false	false	false
"false"	true	false	false	true	false	false	false	false	false		false	false	false	false	false
null	false	true	false	true	false	false	false	false	false	false		false	false	false	true
"foobar"	true	false	false	true	false	false	false	false	false	false	false		false	false	false
"4779"	true	false	false	false	false	false	false	false	false	false	false	false		false	false
"0x12AB"	true	false	false	true	false	false	false	false	false	false	false	false	false		false
""	false	true	false	true	false	false	false	false	false	false	true	false	false	false	

➔ What happens if you expect int but get object?

Linting & Warnings







Lint and Warnings

- Warnings comes from the compiler
- Lint is a seperate tool (ex. Resharper)

- Both help us avoid errors
- ~~Try to~~ **Always** treat warnings as errors.

Error tolerant architectures

Architectural strategies

Handle failures by:

1) Redundancy (think Netflix)
2) Reduced capability

oss lifecycle active | build passing | go reference | go report A

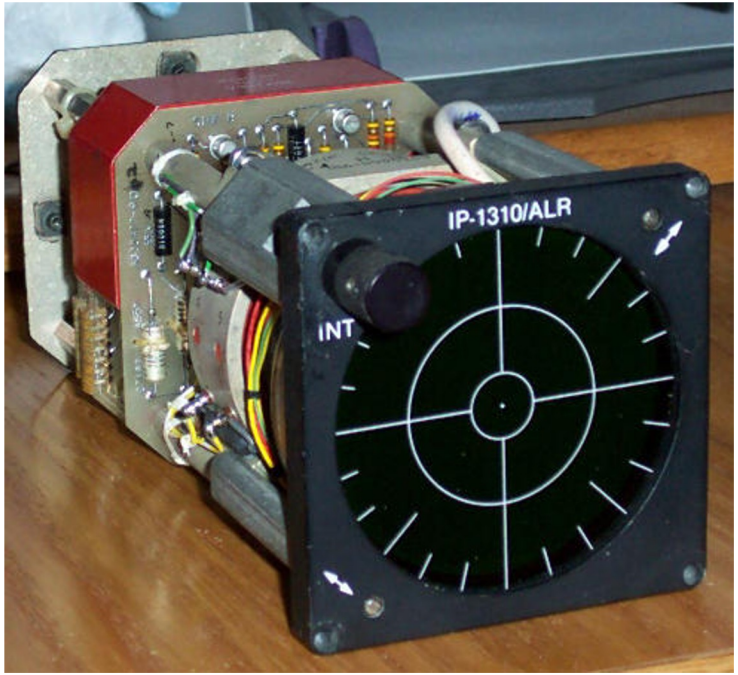
Chaos Monkey randomly terminates virtual machine instances and containers that run inside of your production environment. Exposing engineers to failures more frequently incentivizes them to build resilient services.

See the documentation for info on how to use Chaos Monkey.

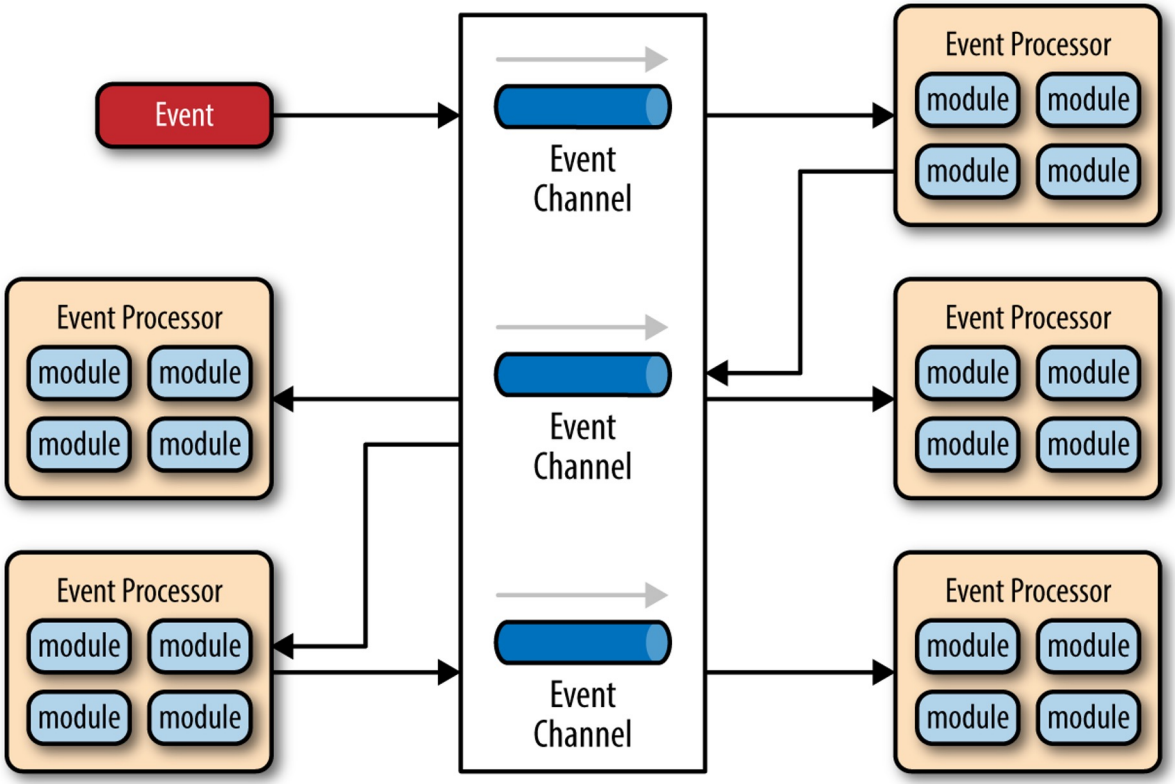
Chaos Monkey is an example of a tool that follows the Principles of Chaos Engineering.

Requirements

Reduced capability



Event driven architecture



Self-testing systems

In military applications:

- Power on Built In Test (PBIT)
- Initiated Built In Test (IBIT)
- Continuous Built In Test (CBIT)

War stories
(aka real-life examples)

War stories

Fail fast

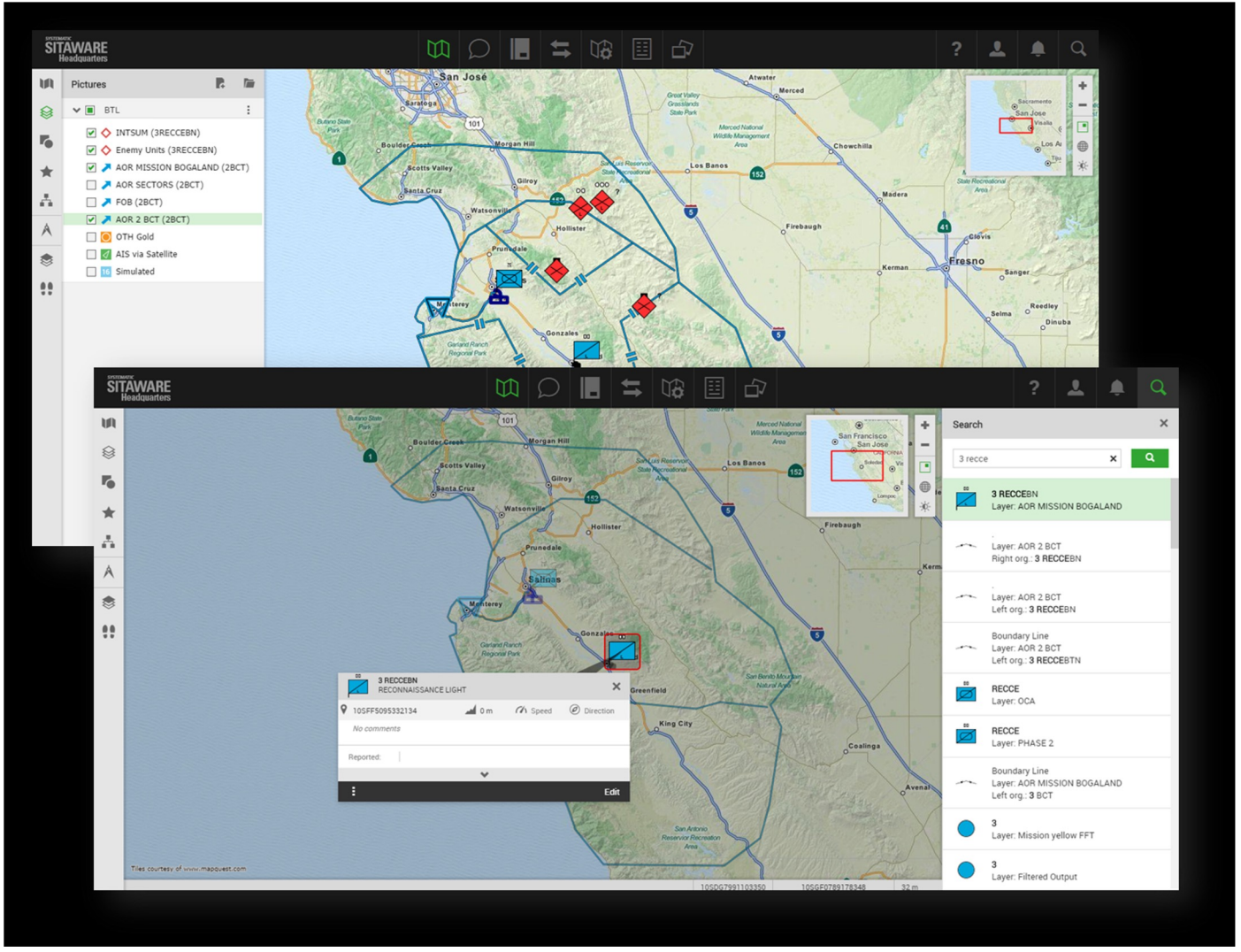
Expect garbage

Never run out of memory

Distributed systems

More garbage

Fail fast

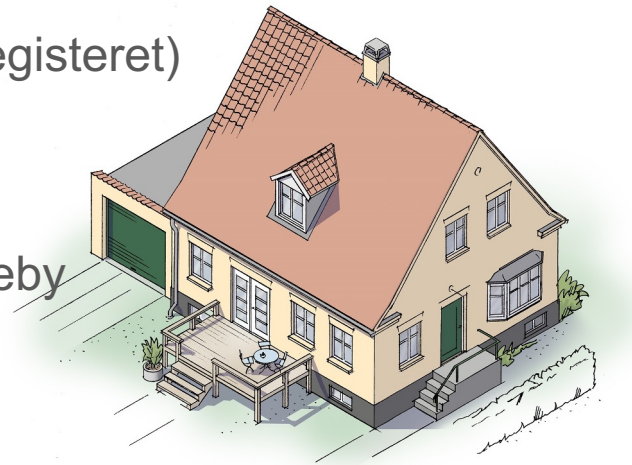

SITAWARE Headquarters
Pictures
BTL
INTSUM (3RECCEBN)
Enemy Units (3RECCEBN)
AOR MISSION BOGALAND (2BCT)
AOR SECTORS (2BCT)
FOB (2BCT)
AOR 2 BCT (2BCT)
OTH Gold
AIS via Satellite
Simulated
San José
Fresno
Search
3 recce
3 RECCEBN
Layer: AOR MISSION BOGALAND
Layer: AOR 2 BCT
Right org.: 3 RECCEBN
Layer: AOR 2 BCT
Left org.: 3 RECCEBN
Boundary Line
Layer: AOR 2 BCT
Left org.: 3 RECCEBTN
RECCE
Layer: OCA
RECCE
Layer: PHASE 2
Boundary Line
Layer: AOR MISSION BOGALAND
Left org.: 3 BCT
3
Layer: Mission yellow FFT
3
Layer: Filtered Output
3 RECCEBN
RECONNAISSANCE LIGHT
10SFF5095332134
No comments
Reported:
Edit
Tiles courtesy of www.mapquest.com

Expect garbage

When you interface to other systems or allow user input, expect external input to be out of specifications. Handle it by sanitizing your input.

Example:

- Interfacing to the BBR register (Bygnings og Bolig-registeret)
  - Missing data in entries
  - Invalid data in entries, e.g. postcode: 9999 Andeby (Ducktown)



What happens if you run out of memory?

function

operator new

<new>

C++98 | C++11

```
throwing (1)    void* operator new (std::size_t size);
nothrow (2)     void* operator new (std::size_t size, const std::nothrow_t& nothrow_value) noexcept;
placement (3)   void* operator new (std::size_t size, void* ptr) noexcept;
```

Allocate storage space

Default *allocation functions* (single-object form).

(1) throwing allocation

Allocates *size* bytes of storage, suitably aligned to represent any object of that size, and returns a non-null pointer to the first byte of this block.
On failure, it throws a `bad_alloc` exception.

(2) nothrow allocation

Same as above *(1)*, except that on failure it returns a *null pointer* instead of throwing an exception.

C++98 | C++11

The default definition allocates memory by calling the the first version: `::operator new (size)`.
If replaced, both the first and second versions shall return pointers with identical properties.

(3) placement

Simply returns *ptr* (no storage is allocated).
Notice though that, if the function is called by a *new-expression*, the proper initialization will be performed (for class objects, this includes calling its default constructor).

What happens if you run out of memory?

How many of you checks for null everytime you create an object?

What happens if you run out of memory?

Strategies:

1) Crash
2) Catch the error and alert the user (if possible)
3) Catch the error and try to continue
4) Never run out of memory

Never run out of memory

In C/C++:

Statically allocate all memory at startup.

New, delete, malloc, free are only allowed in the ‘placement’ versions.

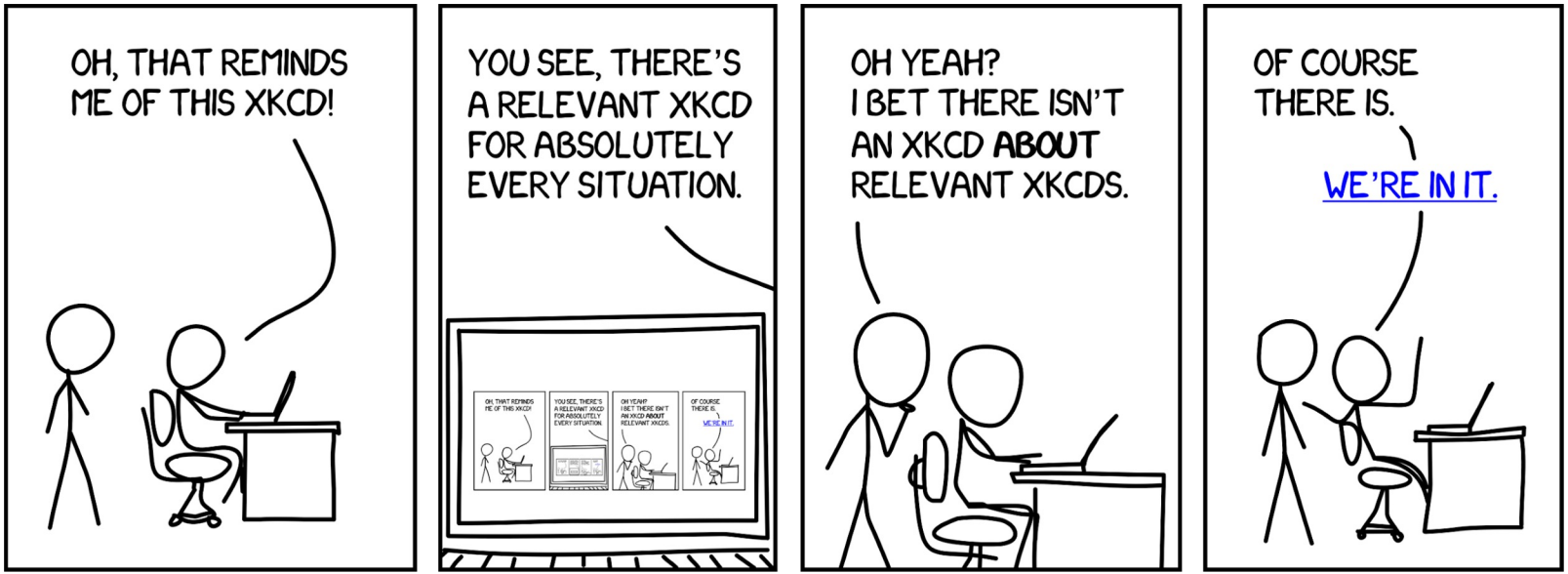
In C#:

?

Never run out of memory

Never run out of memory

Never run out of stack memory



https://cdn-images-1.medium.com/max/2600/1*yNNmaPaMjbto_oSlcO7hvQ.png

Never run out of stack memory

https://cdn-images-1.medium.com/max/2600/1*yNNmaPaMjbto_oSlcO7hvQ.png

Garbage

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

More garbage

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading<h1>
<p>My first paragraph.</p>
</body>
</html>
```

More garbage

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
My First Heading<h1>
<p>My first paragraph.</p>
</body>
</html>
```

More garbage

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
My First Heading<h1>
<p>My first paragraph.</p>
<body>
</html>
```

More garbage

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
My First Heading<h1>
<p>My first paragraph.</p>
<body>
<html>
```

More garbage

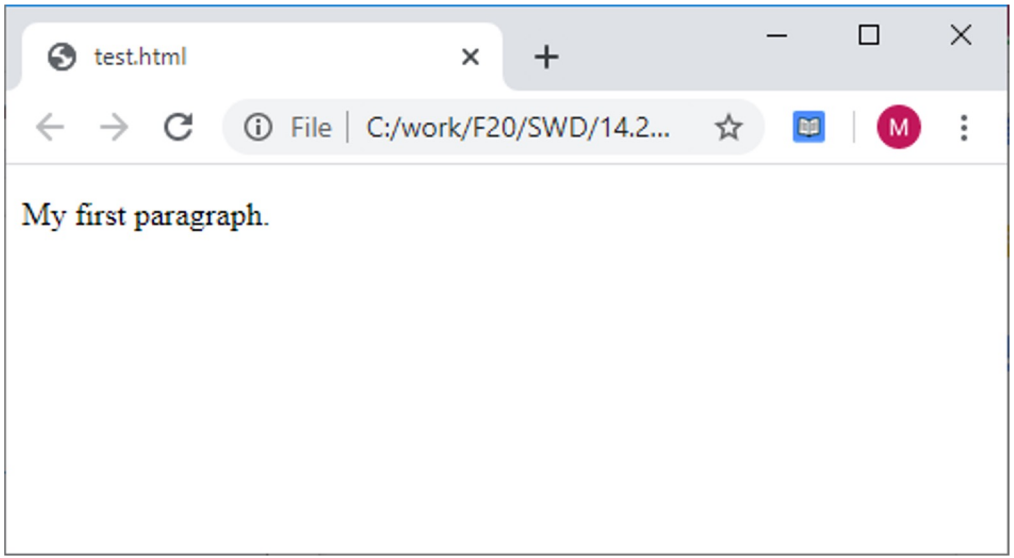
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
My First Heading<h1>
<p>My first paragraph.</p>
<body>
```

More garbage

```
<!DOCTYPE html>

<h1 My First Heading

<p>My first paragraph.</p>
```

More garbage

```
<!DOCTYPE html>

<p>My first paragraph.</p>
```



More garbage

```
<p>My first paragraph.</p>
```

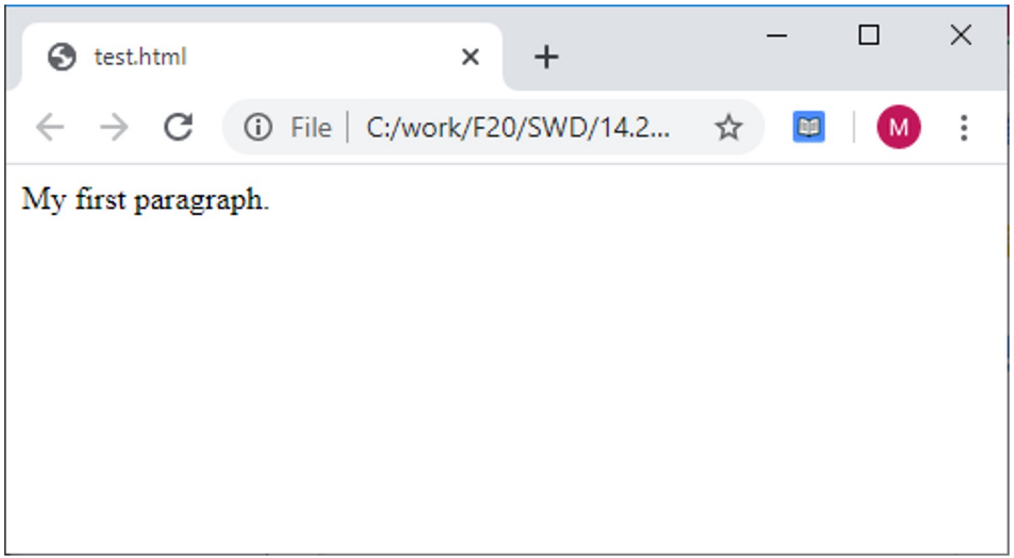


Image references

Velux remote control: http://www.velux.ca/~/media/marketing/us/img/1280/products/skylight-acc-hero.jpg

Terma ACMDS: https://www.terma.com/press/news-2016/terma-raises-the-bar-within-advanced-countermeasures-dispenser-systems/

RWR receiver: https://i.imgur.com/zzo8ZwKh.jpg

RWR receiver: https://www.mikesflightdeck.com/rwr_project/rwr_project.html

Event driven architecture: https://medium.com/@prashunjaveri/architectural-patterns-for-iot-event-driven-architectures-557be35fa626

BBR house: https://bbr.dk/file/654921/bbr-hus.jpeg

SitaWare: https://systematic.com/defence/products/c2/sitaware-headquarters/features/