CNIT 127:
Exploit Development

Lecture 9:
Web Templates and .NET

Not in textbook

New 10-30-19

Web Apps

Legacy Websites

- You can just write HTML files one by one
- The browser loads them with GET requests
- The result is a static, old-fashioned web site, c. 1995

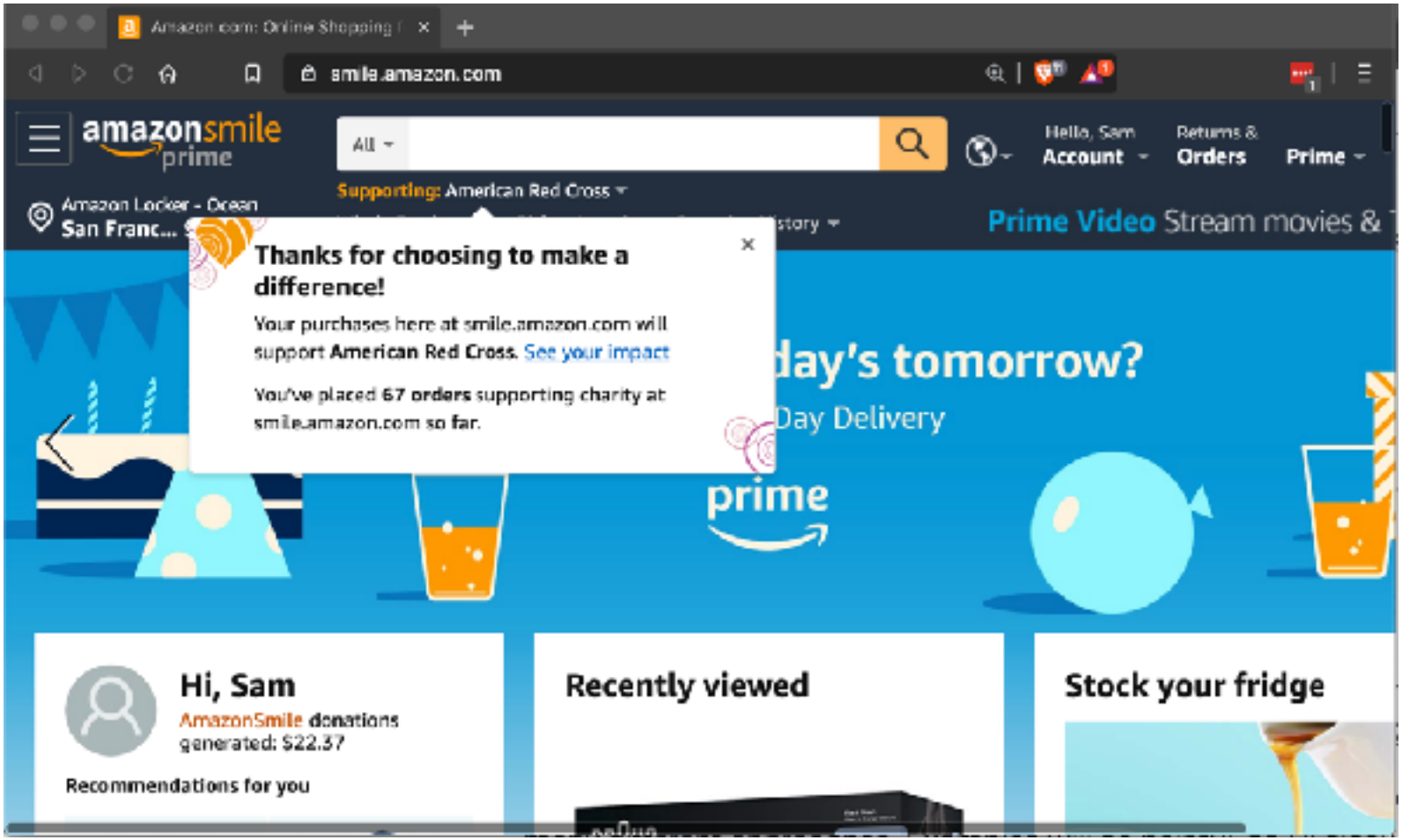
Legacy URL

- Protocol
  - **https**
- Domain
  - https://**samsclass.info**
- Folder name
  - https://samsclass.info/**127**
- File name
  - https://samsclass.info/127/**127_F19.shtml**

Direct Mapping

- The URL directly maps to the actual folders and files on the Web server
- But as business grow and merge, folder names change, and so do domains
- Direct mapping is not flexible or scalable

Modern Website



Web Apps

- Required features
  - POST requests send data to server
  - Sessions and cookies
  - Scale app to thousands of concurrent connections
- Solutions
  - Routing
  - Templates

- Source: https://jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/

Routing

- "the process of mapping a requested URL to the code responsible for generating the associated HTML"
- Simple example:
  - When **www.foo.com/bar** is requested
  - the function **handle_bar()** is used

- Source: https://jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/

Routing in Django

- A request like **www.foo.com/users/3/**
- Use regular expressions
  - URL matching **^/users/(?P<id>\d+)/$**
  - Calls the **display_user(id)**

- Source: https://jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/

Routing in Flask

```
1  @app.route('/users/<id:int>/')
2  def display_user(id):
3      # ...
```

- A request like **www.foo.com/users/3/**

- Source: https://jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/

Routing in ASP DOT NET

C# Copy

```
app.UseMvc(routes =>
{
    routes.MapRoute("default", "{controller=Home}/{action=Index}/{id?}");
});
```

- Matches a URL path like **/Products/Details/5**
- Extracts the route values
  - { controller = **Products**, action = **Details**, id = **5** }

- Source: https://docs.microsoft.com/en-us/aspnet/core/mvc/controllers/routing?view=aspnetcore-3.0

Templates

- Dynamically generate HTML
- Injecting data from URL parameters or from a database

Flask + Jinja2

Structure of Flask App

```
/app
   -/app.py
    /templates
       -/index.html
       -/404.html
```

```
app = Flask(__name__, template_folder='../pages/templates')
```

- From https://codeburst.io/jinja-2-explained-in-5-minutes-88548486834e

Flask + Jinja2

Delimiters

- `{%....%}` are for statements
- `{{....}}` are expressions used to print to template output
- `{#....#}` are for comments which are not included in the template output
- `#....##` are used as line statements

- From https://codeburst.io/jinja-2-explained-in-5-minutes-88548486834e

Flask + Jinja2

Write this file in *app/templates/index.html*:

```
<!DOCTYPE html>
<html>
<p>Hello {{ username }}</p>
</body>
</html>
```

- From https://codeburst.io/jinja-2-explained-in-5-minutes-88548486834e

ASP DOT NET Templates

text

```
└──mytemplate
    │    console.cs
    │    readme.txt
    │
    └──.template.config
         template.json
```

- Source: https://docs.microsoft.com/en-us/dotnet/core/tools/custom-templates

ASP DOT NET Templates

The *template.json* file looks like the following:

JSON

```
{
  "$schema": "http://json.schemastore.org/template",
  "author": "Travis Chau",
  "classifications": [ "Common", "Console" ],
  "identity": "AdatumCorporation.ConsoleTemplate.CSharp",
  "name": "Adatum Corporation Console Application",
  "shortName": "adatumconsole"
}
```

- Source: https://docs.microsoft.com/en-us/dotnet/core/tools/custom-templates

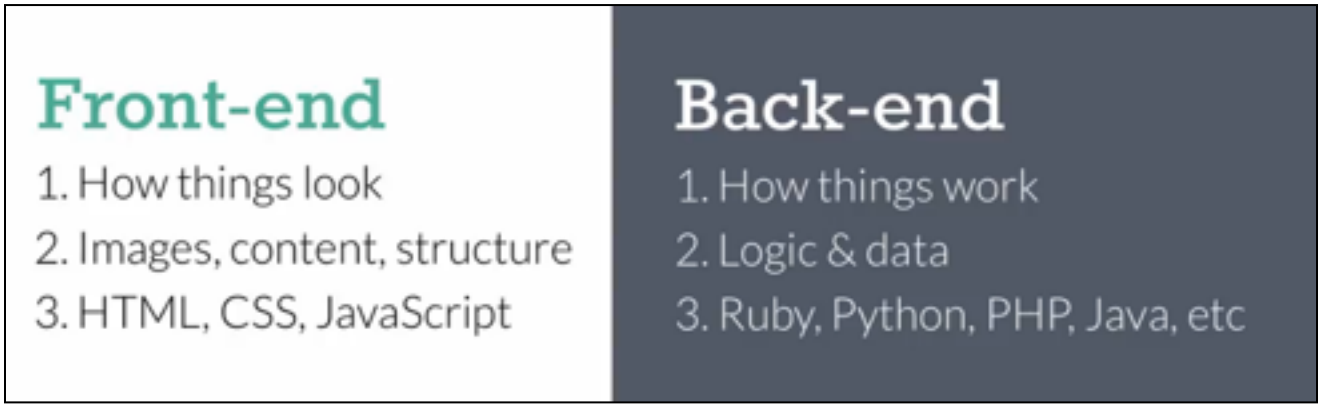
ASP DOT NET Templates

XML Copy

```xml
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <PackageType>Template</PackageType>
    <PackageVersion>1.0</PackageVersion>
    <PackageId>AdatumCorporation.Utility.Templates</PackageId>
    <Title>AdatumCorporation Templates</Title>
    <Authors>Me</Authors>
    <Description>Templates to use when creating an application for Adatum Corporation.</Description>
    <PackageTags>dotnet-new;templates;contoso</PackageTags>
    <TargetFramework>netstandard2.0</TargetFramework>

    <IncludeContentInPack>true</IncludeContentInPack>
    <IncludeBuildOutput>false</IncludeBuildOutput>
    <ContentTargetFolders>content</ContentTargetFolders>
  </PropertyGroup>

  <ItemGroup>
    <Content Include="templates\**\*" Exclude="templates\**\bin\**;templates\**\obj\**" />
    <Compile Remove="**\*" />
  </ItemGroup>

</Project>
```

- Source: https://docs.microsoft.com/en-us/dotnet/core/tools/custom-templates

Front-End v. Back-End



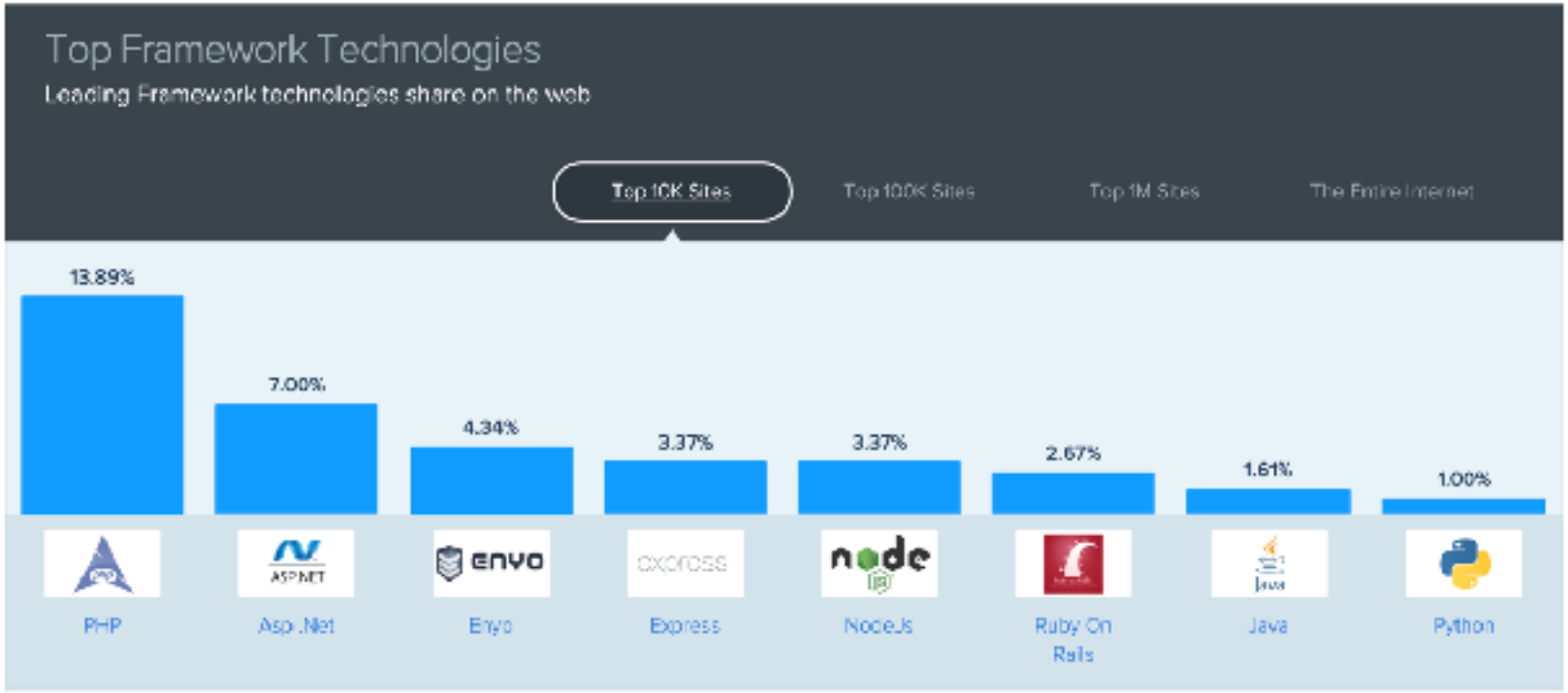
- Source: https://learn.onemonth.com/frontend-vs-backend-developers/

- Source: https://gearheart.io/blog/top-10-web-development-frameworks-2019-2020/

Frontend Frameworks	Pros	Cons
Angular	1. Fast development 2. Good for MVPs and prototyping 3. Good for single-page applications 4. TypeScript support for developing bulky applications 5. Easy to write tests	1. Quite difficult to learn 2. Can have trouble loading huge amounts of data 3. Lacks CLI documentation
Ember	1. Usage of accessors for higher performance 2. Fast development 3. Clear documentation 4. Had its own debugger	1. Difficult to learn 2. Slow rendering 3. Unsuitable for small projects
Flutter	1. Hot Reload 2. Great for MVP 3. Requires less code 4. Cross-platform development	1. Only mobile apps 2. Limited libraries 3. Limited support for Apple TV and Android TV apps
React	1. Virtual DOM for a better experience. 2. Installed components 3. One-direction code flow for a stable code 4. Wide toolset	1. Poor documentation. 2. Slightly verbose, less straightforward than pure JavaScript
Vue.js	1. Small and fast 2. Easy to find errors 3. Clear documentation 4. Simple to integrate with other apps	1. Doesn't have many stable components 2. Far too flexible

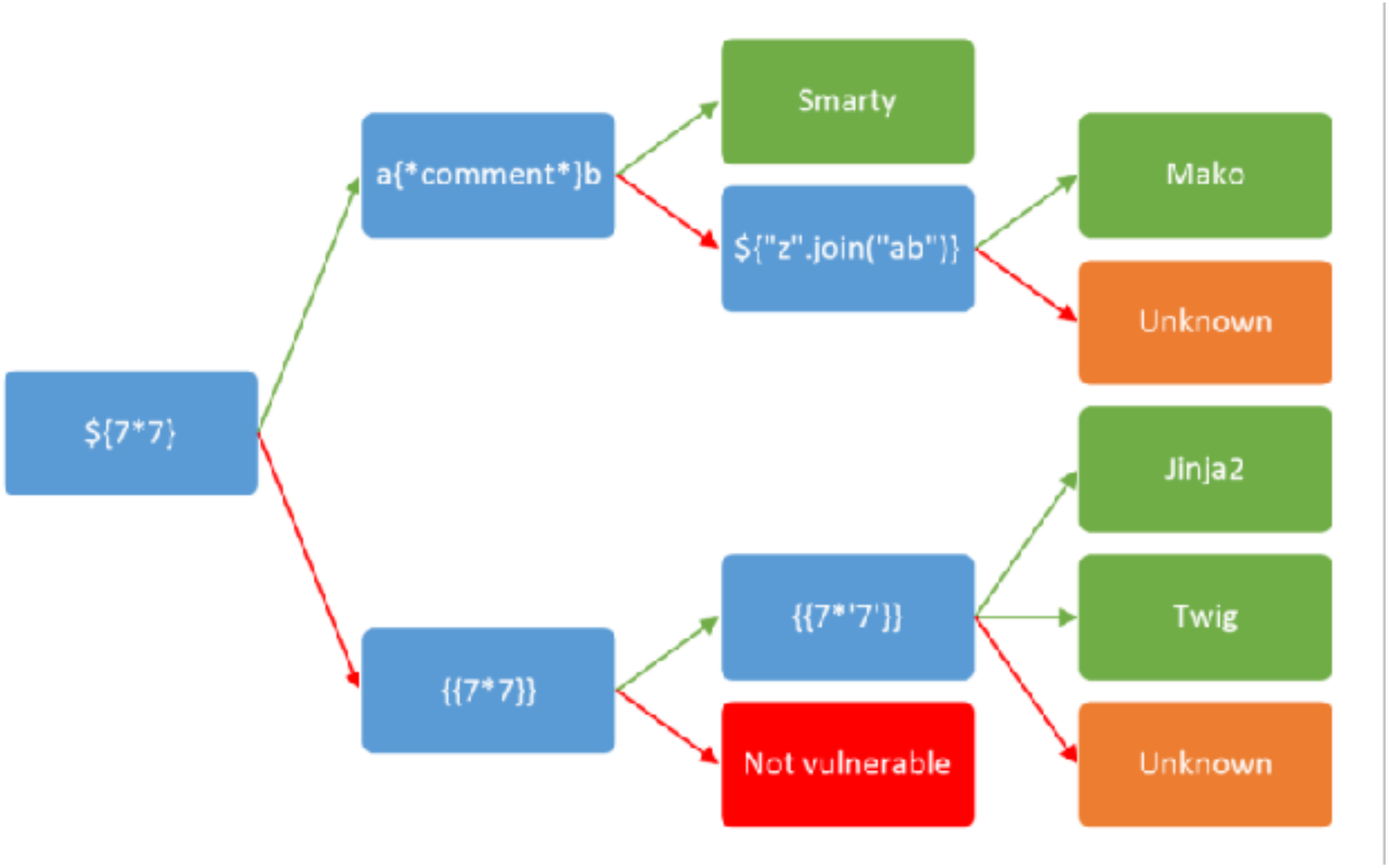
Backend Frameworks	Pros	Cons
Django	1. Scalable and flexible 2. Great for MVPs 3. Secure 4. Great documentation	1. Not the fastest one 2. Monolithic
ExpressJS	1. Simple 2. Flexibility 3. Packages for API development	1. Many callbacks 2. Unhelpful error messages 3. Not suitable for heavy apps
Ruby on Rails	1. Many tools and libraries 2. Fast development 3. Good for prototyping 4. Test automation	1. Slow boot time 2. Not the best choice for heavy applications 3. Lack of proper documentation
Spring	1. Great for Java apps 2. Easy to cooperate with other programs 3. Flexible	1. Difficult to learn 2. Can be unstable
Symfony	1. Fast Development 2. Reusable code 3. Great documentation	1. Comparatively slow

Top Framework Technologies



- Source: https://www.similartech.com/categories/framework

ED 105: Server Side Template Injection (SSTI) (35 pts extra)




Kahoot!

Desktop / Mobile Apps

Types of Executables

- Native code
- Bytecode

Native Code

- Source written in C, C++, Go, etc.
- Compiled to machine language
  - For processor (x86, x64, or ARM)

```
; Attributes: bp-based frame

sub_401160 proc near

var_18= dword ptr -18h
var_14= dword ptr -14h
var_10= byte ptr -10h
var_F= byte ptr -0Fh
var_4= dword ptr -4

push    ebp
mov     ebp, esp
sub     esp, 18h
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax
mov     [ebp+var_18], 0Fh
push    offset aEnterPassword ; "Enter password: "
push    offset unk_434038
call    sub_4015F0
add     esp, 8
lea     eax, [ebp+var_10]
push    eax
push    offset dword_433FC0
call    sub_401350
add     esp, 8
mov     [ebp+var_14], 0
jmp     short loc_4011AC
```

Bytecode and IL

- Source written in Java
  - Compiled to bytecode
  - Platform-independent
  - Runs in a Java Virtual Machine
- Source written in C# or Visual Basic, or many other languges
  - Compiled by the Common Language Runtime (CLR) to Intermediate Language (IL)
  - Executed by the .NET CLR "Just-In-Time compiler"
  - Windows only, in practice

Java Source Code

```
outer:
for (int i = 2; i < 1000; i++) {
    for (int j = 2; j < i; j++) {
        if (i % j == 0)
            continue outer;
    }
    System.out.println (i);
}
```

- Source: https://en.wikipedia.org/wiki/Java_bytecode

Java Bytecode

```
0:   iconst_2
1:   istore_1
2:   iload_1
3:   sipush  1000
6:   if_icmpge       44
9:   iconst_2
10:  istore_2
11:  iload_2
12:  iload_1
13:  if_icmpge       31
16:  iload_1
17:  iload_2
18:  irem
19:  ifne    25
22:  goto    38
25:  iinc    2, 1
28:  goto    11
31:  getstatic       #84; // Field java/lang/System.out:Ljava/io/PrintStream;
34:  iload_1
35:  invokevirtual   #85; // Method java/io/PrintStream.println:(I)V
38:  iinc    1, 1
41:  goto    2
44:  return
```

- Source: https://en.wikipedia.org/wiki/Java_bytecode

C# Source Code

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "topsecret")
        MessageBox.Show("WIN!");
    else
        MessageBox.Show("FAIL!");
}
```

.NET Intermediate Language (IL)

```
.method private hidebysig instance void button1_Click(object sender, class [mscorlib]System.EventArgs e) cil managed
{
  .maxstack 2
  .locals init (
    [0] bool flag)
  L_0000: nop
  L_0001: ldarg.0
  L_0002: ldfld class [System.Windows.Forms]System.Windows.Forms.TextBox WindowsFormsApp5.Form1::textBox1
  L_0007: callvirt instance string [System.Windows.Forms]System.Windows.Forms.Control::get_Text()
  L_000c: ldstr "topsecret"
  L_0011: call bool [mscorlib]System.String::op_Equality(string, string)
  L_0016: stloc.0
  L_0017: ldloc.0
  L_0018: brfalse.s L_0027
  L_001a: ldstr "WIN!"
  L_001f: call valuetype [System.Windows.Forms]System.Windows.Forms.DialogResult [System.Windows.Forms]System.Windows.Forms.MessageBox::Show(string)
  L_0024: pop
  L_0025: br.s L_0032
  L_0027: ldstr "FAIL!"
  L_002c: call valuetype [System.Windows.Forms]System.Windows.Forms.DialogResult [System.Windows.Forms]System.Windows.Forms.MessageBox::Show(string)
  L_0031: pop
  L_0032: ret
}
```

Feature	Microsoft .NET	Java
Programming Languages	C#, VB.NET, C++, .NET, PHP, Ruby, Python & more	Java, Clojure, Groovy, Scala, PHP, Ruby, Python, JavaScript & more
Operating System	Windows	Multiple
Runtime	CLR	JVM
Server Components	.NET, COM + Serviced	EJBs
GUI Components	.NET Class	Java Beans
Web Services Support	Built-in	Add-on
Unit Testing	Microsoft Unit Testing Framework, NUnit	JUnit
Web Application Framework	ASP.NET MVC, Spring .NET	Spring
Web Server Scripting	ASP.NET	JSF
Data Access	ADO.NET / oLeDB	JDBC
HTTP Engine	IIS	Application Servers from Multiple Vendors
Remoting	SOAP, HTTP, DCOM	RMI-over-IIOP

- Source: https://www.rishabhsoft.com/blog/dot-net-vs-java

Choose Java if:

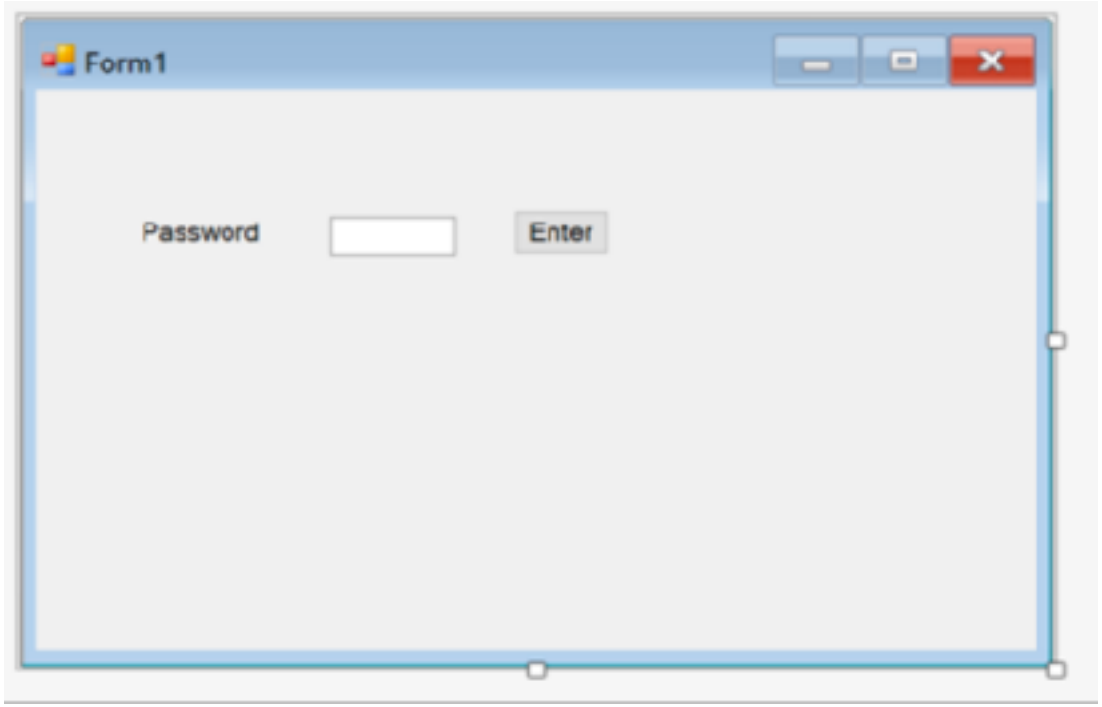
- You want an enterprise-grade application
- Seeking Portability & Platform independence
- Your application attracts high user volumes

Choose .NET if:

- Develop web services
- Highly secure application
- Feature-packed, intuitive application with a rich GUI

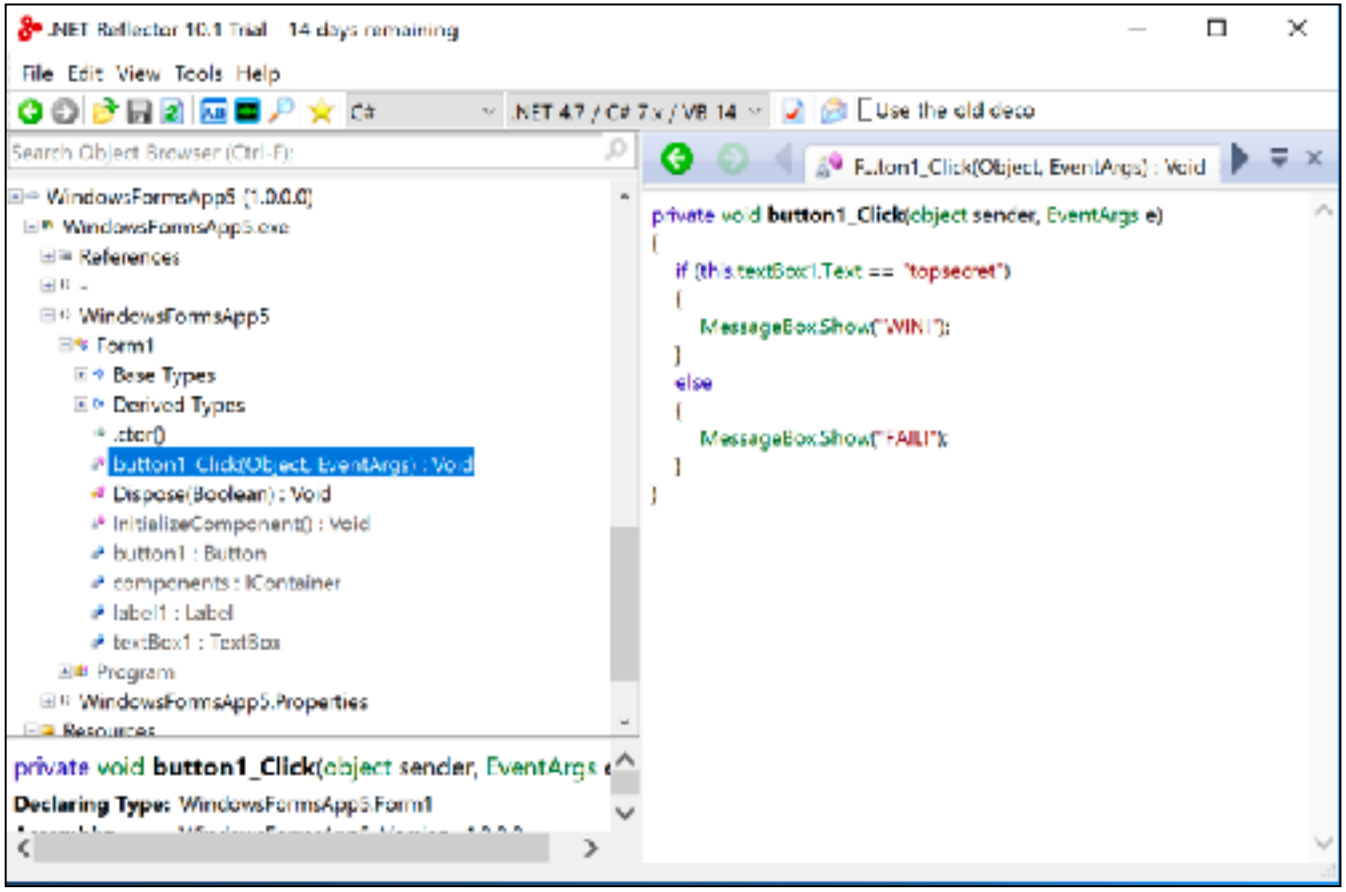
- Source: https://www.rishabhsoft.com/blog/dot-net-vs-java

ED 330: C# Dot Net (20 pts extra)




```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "topsecret")
        MessageBox.Show("WIN!");
    else
        MessageBox.Show("FAIL!");
}
```

ED 331: Dot Net Reflector (45 pts extra)




Kahoot!