

試験にでないIT雑学

第2回OTM(OAS Tech Meeting)

はじめに

- 今回は、知らなくてもあまり困らない（かもしれない）、ちょっとした情報をお伝えします。
他の方の発表のように、特化した技術情報ではないので、いろいろな情報が混ざっています。
「全部知っているよ」という情報かもしれませんが、しばらくお付き合いください。

試験にでないIT雑学①

試験にでないIT雑学①

- マジックナンバーとは…？

A

魔法の
数字

B

優勝するまで
の勝利の数

C

フォーマット
識別子

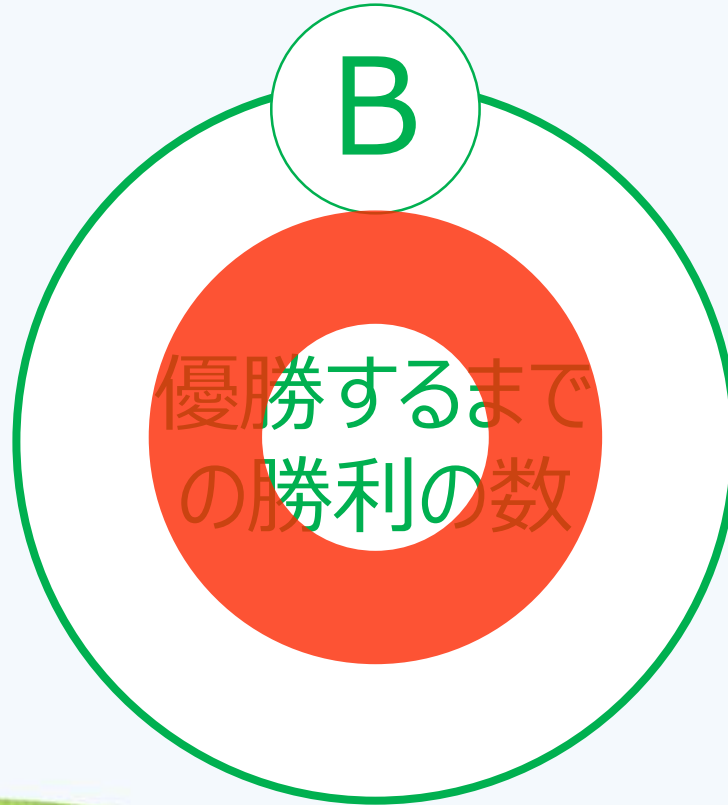
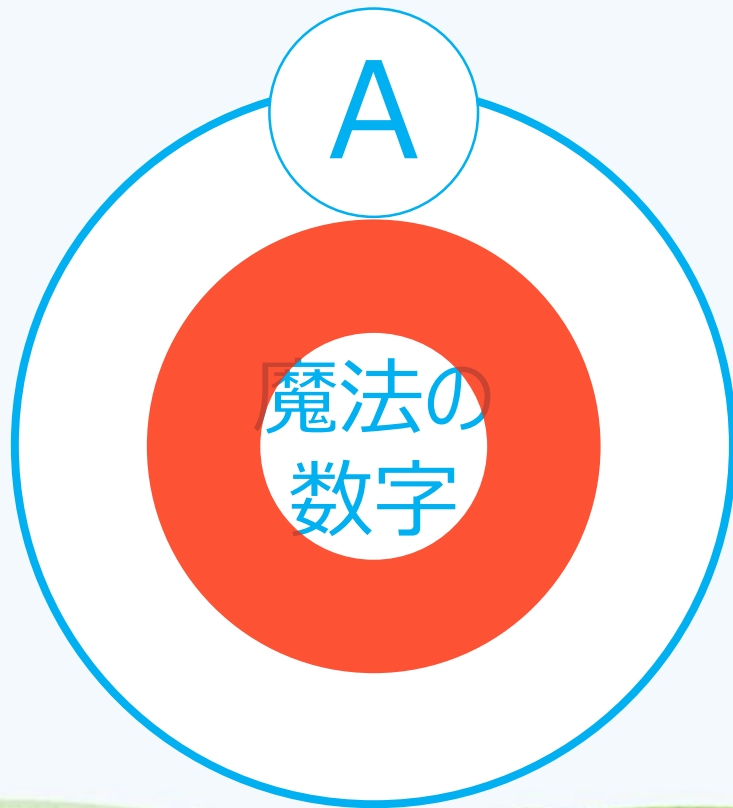
試験にでないIT雑学①

- マジックナンバーとは…？



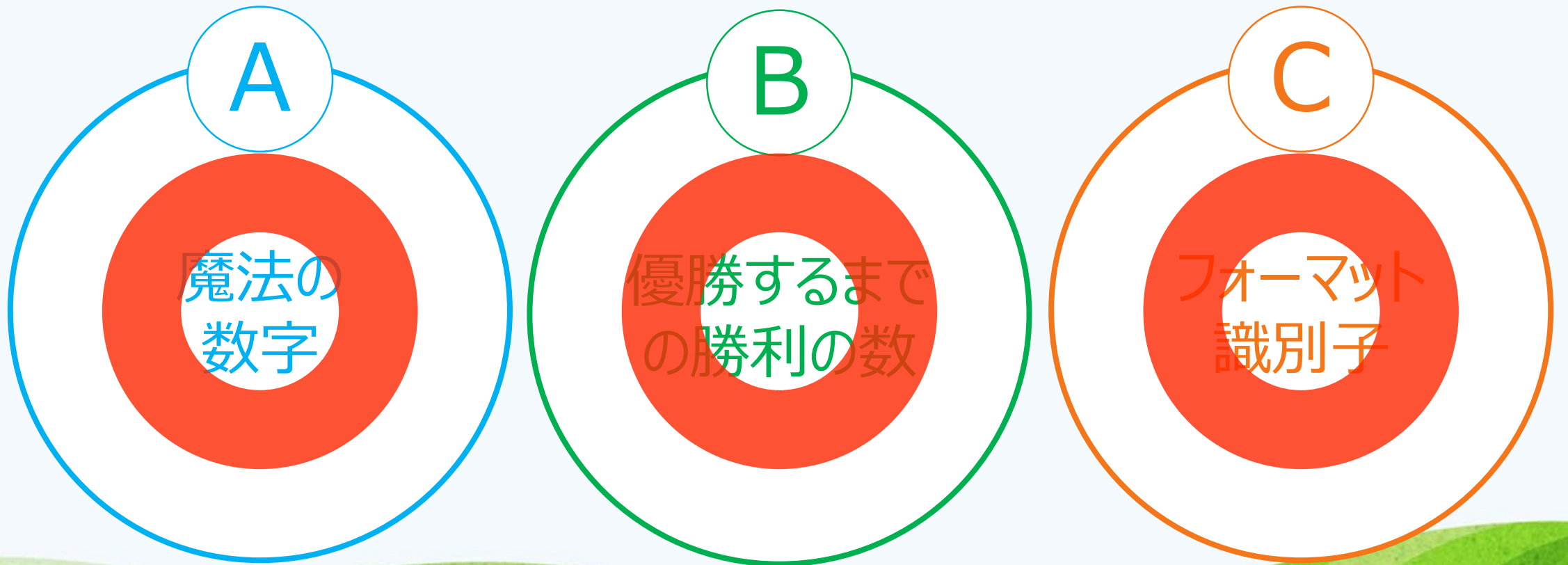
試験にでないIT雑学①

- マジックナンバーとは…？



試験にでないIT雑学①

- マジックナンバーとは…？



試験にでないIT雑学①

- マジックナンバーとは…？



試験にでないIT雑学①

- マジックナンバーとは…？

A

魔法の
数字

B

優勝するまで
の勝利の数

C

フォーマット
識別子

試験にでないIT雑学①

A

魔法の
数字

- ソースコードの中に突然現れる、書いた本人しか意味がわからない数字のことです。（本人も意味がわからない時も…）仕様書もコメントもなく引き継いだ場合、魔法ではなく悪魔の数字となります。
 - `if (categoryCode == 200)`

試験にでないIT雑学①

B

優勝するまで
の勝利の数

- プロ野球で、チームが優勝するまでに、あと何勝すれば優勝できるかを表す数字のことです。
 - ITとは関係ありません…すいません…
 - ちなみに…
昨年のヤクルトスワローズは9月27日に、マジック3が点灯しました

試験にでないIT雑学①

C

フォーマット
識別子

- ファイルの種類を識別するために、ファイルの中の特定の位置に埋め込まれた数字のことです。
 - 例えば、先頭の4バイトが
50 4B 03 04
の場合、ZIPファイルとなります。

試験にでないIT雑学②

試験にでないIT雑学②

- Javaのクラスファイルのマジックナンバーは、どれでしょうか？

A

CA FE BA
BE

B

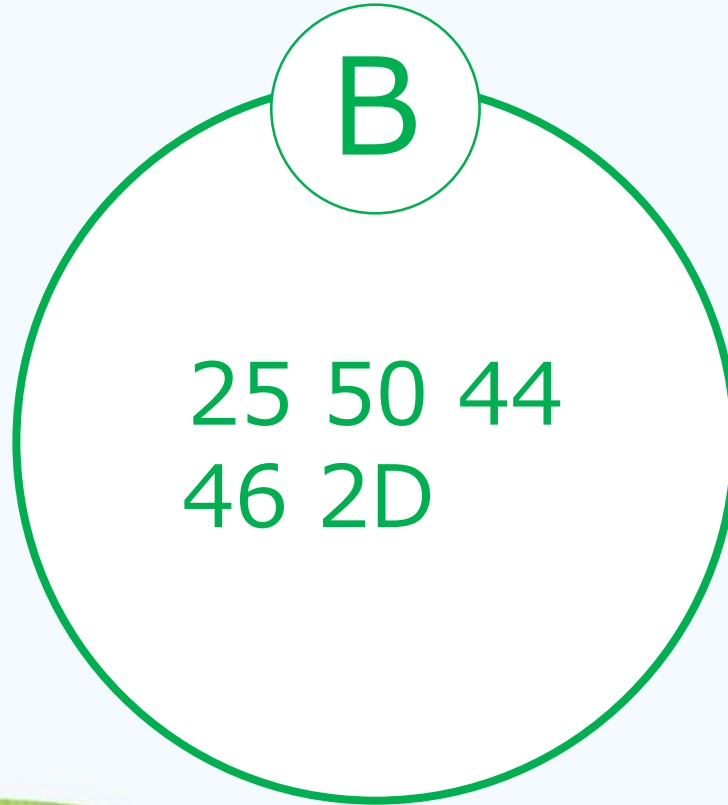
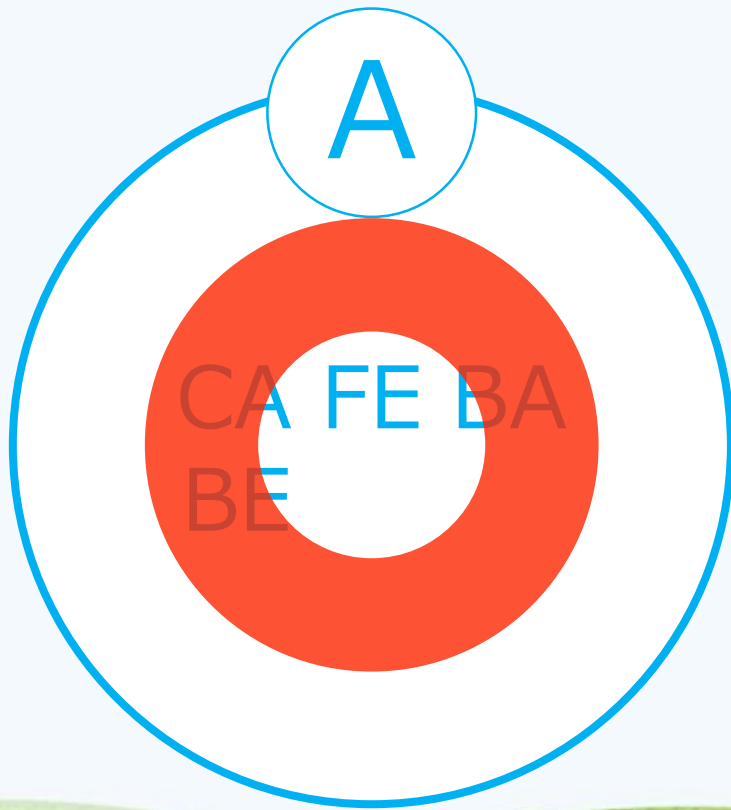
25 50 44
46 2D

C

C0 FF EE

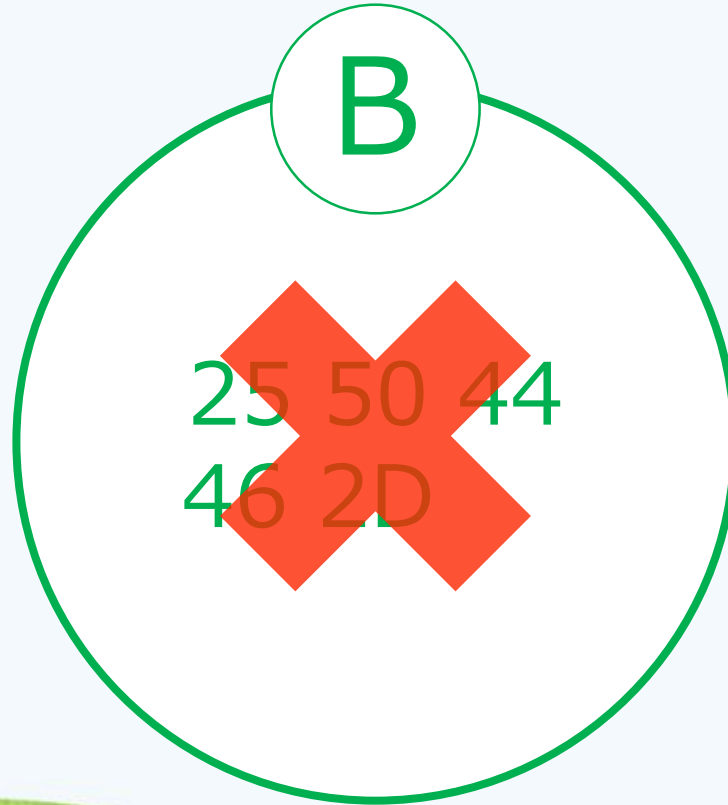
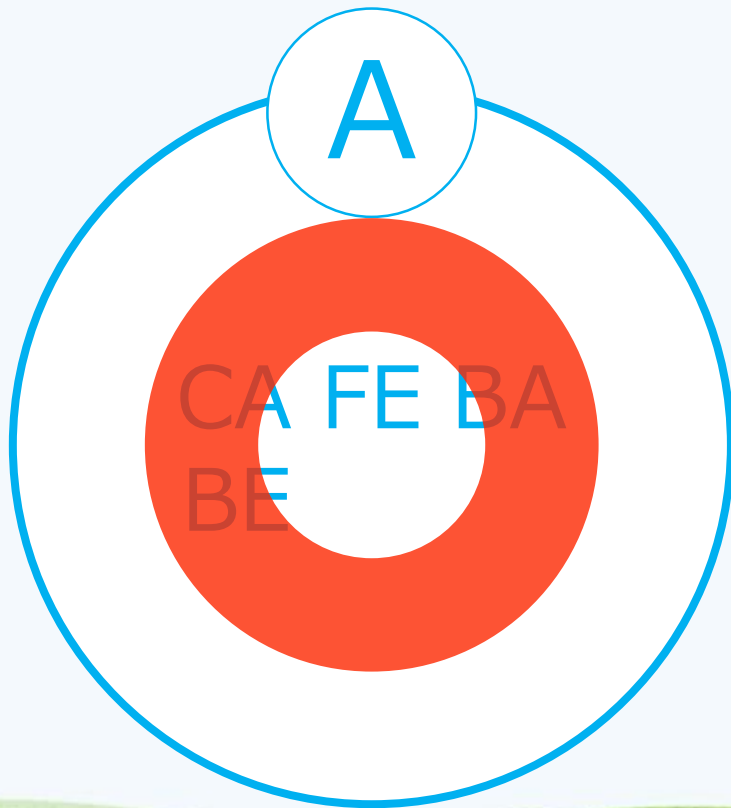
試験にでないIT雑学②

- Javaのクラスファイルのマジックナンバーは、どれでしょうか？



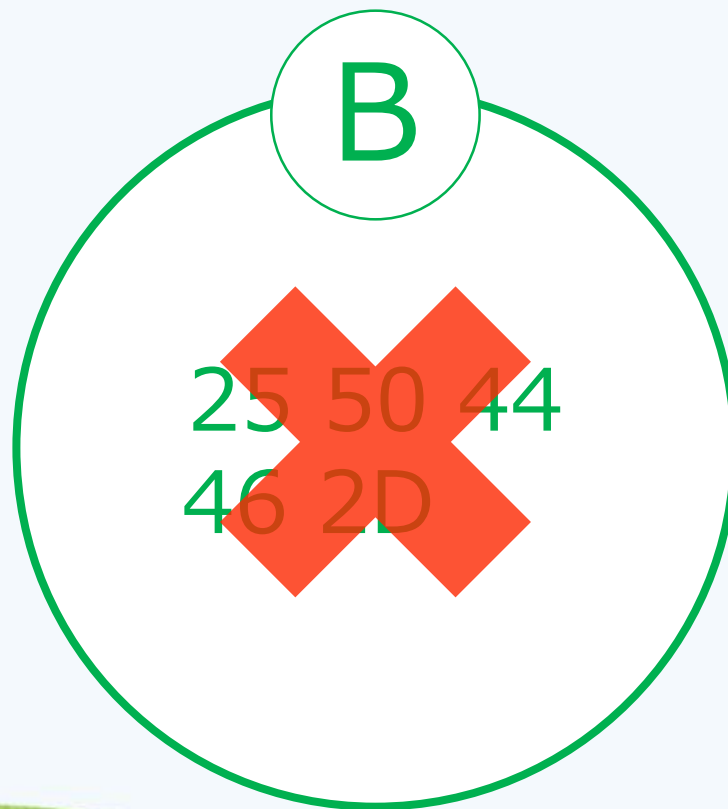
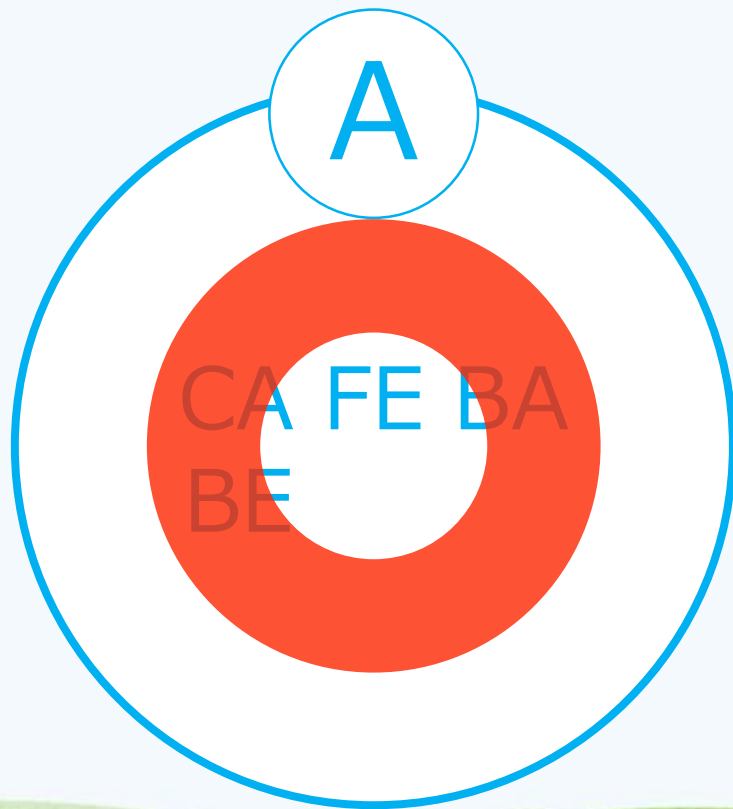
試験にでないIT雑学②

- Javaのクラスファイルのマジックナンバーは、どれでしょうか？



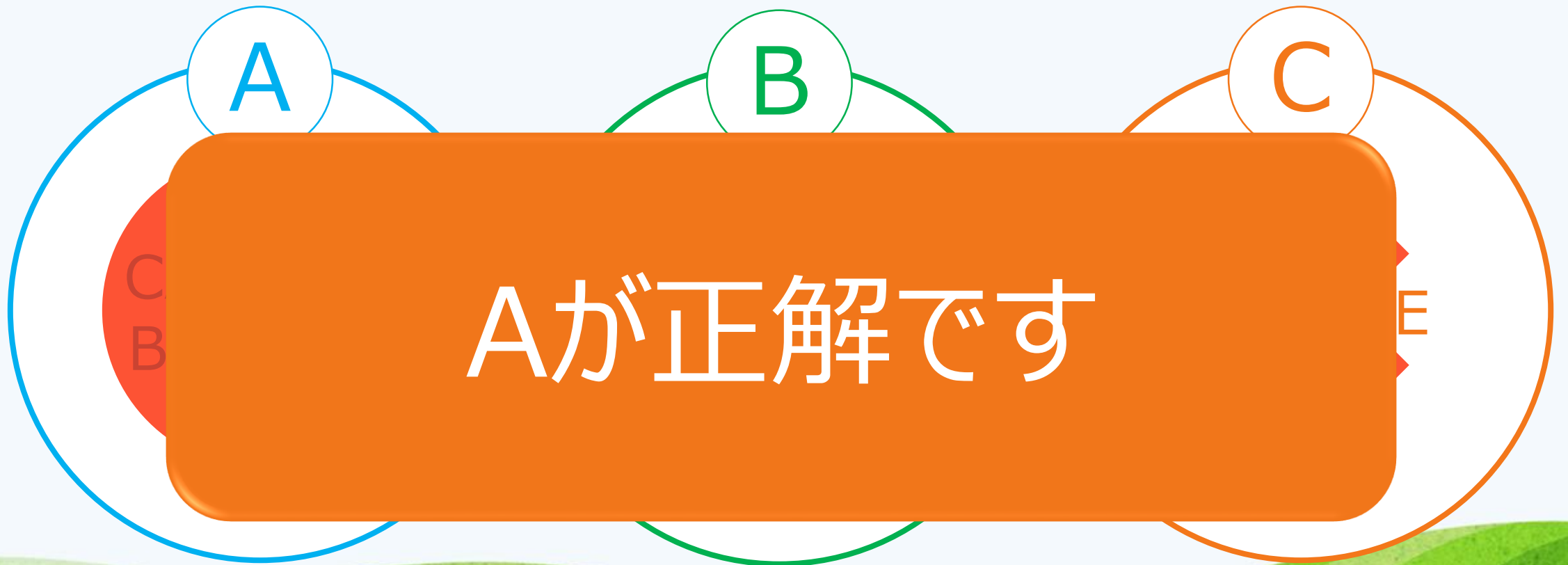
試験にでないIT雑学②

- Javaのクラスファイルのマジックナンバーは、どれでしょうか？



試験にでないIT雑学②

- Javaのクラスファイルのマジックナンバーは、どれでしょうか？



試験にでないIT雑学②

- Javaのクラスファイルのマジックナンバーは、どれでしょうか？

A

CA FE BA
BE

B

25 50 44
46 2D

C

C0 FF EE

試験にでないIT雑学②

A

CA FE BA
BE

- Javaのクラスファイルをバイナリで覗いてみると、以下のようになっています。

```
d:¥otm>certutil -encodehex Q1.class Q1_class.txt 11
```

```
d:¥otm>type Q1_class.txt
```

```
0000  ca fe ba be 00 00 00 34  00 1d 0a 00 06 00 0f 09  .....4.....
```

```
0010  00 10 00 11 08 00 12 0a  00 13 00 14 07 00 15 07  .....
```


試験にでないIT雑学②

A

CA FE BA
BE

- Javaのクラスファイルをバイナリで覗いてみると、以下のようになってい

ca fe ba be

.class Q1_class.txt 11

```
0000 ca fe ba be 00 00 00 34 00 1d 0a 00 06 00 0f 09 .....4.....
0010 00 10 00 11 08 00 12 0a 00 13 00 14 07 00 15 07 .....
```

試験にでないIT雑学②

- 参考情報：Javaのクラスファイルの構成

```
ClassFile {  
    u4          magic;  
    u2          minor_version;  
    u2          major_version;  
    u2          constant_pool_count;  
    cp_info     constant_pool[constant_pool_count-1];  
    u2          access_flags;  
    u2          this_class;  
    u2          super_class;  
    u2          interfaces_count;  
    u2          interfaces[interfaces_count];  
    u2          fields_count;  
    field_info  fields[fields_count];  
    u2          methods_count;  
    method_info methods[methods_count];  
    u2          attributes_count;  
    attribute_info attributes[attributes_count];  
}
```

<https://docs.oracle.com/javase/specs/jvms/se8/html/jvms-4.html#jvms-4.1>

試験にでないIT雑学②

- 参考情報：Javaのクラスファイルの構成

```
ClassFile {  
    u4          magic;  
    u2          minor_version;  
    u2          ...  
    u2          ...  
    cp_info     ...  
    u2          ...  
    u2          ...  
    u2          ...  
    u2          interfaces_count;  
    u2          interfaces[interfaces_count];  
    u2          fields_count;  
    field_info  fields[fields_count];  
    u2          methods_count;  
    method_info methods[methods_count];  
    u2          attributes_count;  
    attribute_info attributes[attributes_count];  
}
```

先頭4バイトがマジックナンバー

<https://docs.oracle.com/javase/specs/jvms/se8/html/jvms-4.html#jvms-4.1>

試験にでないIT雑学②

B

25 50 44
46 2D

- これは、PDFファイルのマジックナンバーです。
PDFファイルをバイナリで覗いてみると、以下のようになっています。

```
D:¥otm>type Q1_pdf.txt
```

```
0000 25 50 44 46 2d 31 2e 34 0a 25 e2 e3 cf d3 0a 33 %PDF-1.4.%.....3
```

```
0010 20 30 20 6f 62 6a 0a 3c 3c 2f 45 20 32 33 39 32 0 obj.<</E 2392
```

試験にでないIT雑学②

B

25 50 44
46 2D

- これは、PDFファイルのマジックナンバーです。
PDFファイルを開いてみると、以下のようになっている

25 50 44 46 2d

```
0000 25 50 44 46 2d 31 2e 34 0a 25 e2 e3 cf d3 0a 33 %PDF-1.4.%.....3
0010 20 30 20 6f 62 6a 0a 3c 3c 2f 45 20 32 33 39 32 0 obj.<</E 2392
```

試験にでないIT雑学②

C

CO FF EE

- 何のファイルのマジックナンバーでもありません…

試験にでないIT雑学③

試験にでないIT雑学③

- PowerShellを使ったことがありますか？

A

使ったことある

B

使ったことない

C

PowerShell
って何？

試験にでないIT雑学③

C

PowerShell
って何？

- PowerShellは、ざっくり説明すると、コマンドプロンプトの進化形です。
.Net Frameworkの上で動作しています。

試験にでないIT雑学③

C

PowerShell
って何？

- Windows 7以降は標準で搭載されています。
 - Windows 7 => PowerShell 2.0
 - Windows 8 => PowerShell 3.0
 - Windows 8.1 => PowerShell 4.0
 - Windows 10 => PowerShell 5.0

試験にでないIT雑学③

B

使ったことない

C

PowerShell
って何？

- 例えば、現在実行中のサービスの一覧を取得したい場合、以下のようなコマンドを実行することにより取得することができます。

```
PS D:\otm> Get-Service | ? {$_.Status -eq "Running"}
```

Status	Name	DisplayName
-----	----	-----
Running	AdobeARMservice	Adobe Acrobat Update Service

試験にでないIT雑学③

B

使ったことない

C

PowerShell
って何？

- スクリプトを作成して実行することもできます。
例えば、次ページのスクリプトは、指定されたフォルダの容量をGB/MB/B単位で表示するスクリプトです。

試験にでないIT雑学③

```
Param (
```

```
$TargetFolder = "C:¥Program Files"
```

```
)
```

```
$Items = (Get-ChildItem -Path $TargetFolder -Recurse -ErrorAction "SilentlyContinue" | ?  
{$_ .GetType().Name -eq "FileInfo"} | Measure-Object Length -sum)
```

```
[Array]$Folders = [PSCustomObject]@{
```

```
    "Total Size(GB)" = ("{$0:N2}" -f ($Items.Sum / 1024000000))
```

```
    "Total Size(MB)" = ("{$0:N2}" -f ($Items.Sum / 1024000))
```

```
    "Total Size(B)" = ("{$0:N0}" -f $Items.Sum)
```

```
}
```

```
$Folders | Format-Table -AutoSize
```

試験にでないIT雑学③

A

使ったことある

B

使ったことない

C

PowerShell
って何？

- PowerShell 5.0では、classも作成できるようになっています。
いろいろな言語で有名な、「Hello World」を、PowerShellで作成すると、次ページのようになります。

試験にでないIT雑学③

```
function Main() {  
    $instance = [HelloWorld]::New("Hello World.");  
    $instance.Print();  
}  
class HelloWorld {  
    [string]$Word;  
  
    HelloWorld([string]$word) {  
        $this.Word = $word;  
    }  
    [void] Print() {  
        [Console]::WriteLine($this.Word);  
    }  
}
```

試験にでないIT雑学④

試験にでないIT雑学④

- 以下のJavaソースコードを実行すると、どんな結果になるでしょうか？

```
public class Otm2Q4 {  
    public static final Otm2Q4 INSTANCE = new Otm2Q4();  
    private static final String CONST_STR = new String("OTM2_Q4");  
    private final String RESULT_STR;  
    private Otm2Q4() { RESULT_STR = String.format("[%s]", Otm2Q4.CONST_STR); }  
    public String getResult() { return this.RESULT_STR; }  
    public static void main(String[] args) {  
        System.out.println(INSTANCE.getResult());  
    }  
}
```

試験にでないIT雑学④

- 以下のJavaソースコードを実行すると、どんな結果になるでしょうか？

```
public class Otm2Q4 {
```

A

[OTM2_Q4]

B

[null]

C

[]

試験にでないIT雑学④

- 以下のJavaソースコードを実行すると、どんな結果になるでしょうか？

```
public class Otm2Q4 {
```

A

[OTM2_Q4]



B

[null]

C

[]

試験にでないIT雑学④

- 以下のJavaソースコードを実行すると、どんな結果になるでしょうか？

```
public class Otm2Q4 {
```

A

[OTM2_Q4]

B

[null]

C

[]

試験にでないIT雑学④

- 以下のJavaソースコードを実行すると、どんな結果になるでしょうか？

```
public class Otm2Q4 {
```

A

[OTM2_Q4]

B

[null]

C

[]

試験にでないIT雑学④

- 以下のJavaソースコードを実行すると、どんな結果になるでしょうか？

```
public class Otm2Q4 {  
    public static final Otm2Q4 INSTANCE = new Otm2Q4();  
    private final String CONST = "OTM2_Q4";  
}
```

A

B

C

[O

Bが正解です

試験にでないIT雑学④

B

[null]

- "[null]"となる理由は、
mainメソッドが実行されるまでに、何が、どのような順番
で実行されているのか？
ということが重要です。

試験にでないIT雑学④

- mainメソッドが実行されるまでに、実行される処理は、いろいろとありますが、今回のソースコードに関係のある部分を抜粋すると、以下のようになります。
 - ① staticフィールドを初期化します。(Javaの処理は実行されません)
 - ② Javaの処理を実行してstaticフィールドを初期化します。
 - ③ mainメソッドを実行します

試験にでないIT雑学④

- 「①」の時点では、staticフィールドはJavaの処理が実行されず、初期化だけが行われていますので、
 INSTANCE = null
 CONST_STR = null
となっています。

試験にでないIT雑学④

- 「②」の時点では、Javaの処理が実行されて、初期化が行われますので、最初に
 INSTANCE = new Otm2Q4()
が実行されます。
 コンストラクタの中で、CONST_STRを参照していますが、Javaの処理は上から順番に実行されますので、
 CONST_STR = new String("OTM2_Q4")
は、まだ実行されていません。
 そのため、[null]と表示されることになります。

試験にでないIT雑学⑤

試験にでないIT雑学⑤

- 以下のJavaScriptを実行すると、どんな結果になるでしょうか？

```
var array = [1, 2, 3];  
array[100] = 100;  
array['end'] = 200;  
var ans1 = 0;  
var ans2 = 0;  
for (var i = 0, len = array.length; i < len; i++) {  
    ans1++;  
}  
for (var key in array) {  
    ans2++;  
}  
alert(ans1 + "::" + ans2);
```

試験にでないIT雑学⑤

- 以下のJavaScriptを実行すると、どんな結果になるでしょうか？

```
var array = [1, 2, 3];
```

```
array[0] = 100;
```

```
array[1] = 200;
```

A

4::5

B

5::5

C

101::5

```
array[2] = array[0] + array[1];
```

試験にでないIT雑学⑤

- 以下のJavaScriptを実行すると、どんな結果になるでしょうか？

```
var array = [1, 2, 3];
```

```
array[0] = 100;
```

```
array[1] = 200;
```

A



4::5

B

5::5

C

101::5

```
array[2] = array[0] + array[1];
```


試験にでないIT雑学⑤

- 以下のJavaScriptを実行すると、どんな結果になるでしょうか？

```
var array = [1, 2, 3];
```

```
array[100] = 100;
```

```
array[101] = 200;
```

A



4::5

B



5::5

C

101::5

```
array[100] + array[101];
```

試験にでないIT雑学⑤

- 以下のJavaScriptを実行すると、どんな結果になるでしょうか？

```
var array = [1, 2, 3];
```

```
array[5] = 100;
```

```
array[5] = 200;
```

A



B



C

101::5



```
array[5] + ans2);
```

試験にでないIT雑学⑤

- 以下のJavaScriptを実行すると、どんな結果になるでしょうか？

```
var array = [1, 2, 3];
```

```
array[0] = 100;
```

```
array[1] = 200;
```

A

B

C

Cが正解です

```
array[2] = array[0] + ans2);
```

試験にでないIT雑学⑤

C

101::5

- JavaScriptの配列のlengthプロパティは、配列の有効なindex値(整数値)に1が加算された値が格納されています。
そのため、
 `array[100] = 100;`
の時に、lengthプロパティは101になりますが、その後の、
 `array['end'] = 200;`
の'end'は有効なindex値ではないため、lengthプロパティは変更されません。