

## Une démo de Marp et de la customisation de thème

Testing purpose



email : [martial.braux@gmail.com](mailto:martial.braux@gmail.com)

Twitter : <https://twitter.com/mbraux>

LinkedIn : <https://www.linkedin.com/in/mbraux/>

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc convallis urna turpis, ut accumsan enim aliquet vitae. In vel ex fringilla, porttitor turpis vel, suscipit dui. Nunc ante quam, ullamcorper ut hendrerit eu, luctus eu turpis. Nam pretium tempor libero a tempor.

## et cetera

Etiam rutrum placerat molestie. Duis eget lacus nibh. Nam scelerisque purus sit amet condimentum interdum. Phasellus sit amet hendrerit neque, id maximus leo.

## Alea jacta est

Nulla tincidunt eu nisi a pulvinar. Quisque semper rutrum consequat. Pellentesque quis volutpat justo, in pellentesque enim. Praesent gravida consequat scelerisque. Praesent vulputate eu massa eget tempus. Proin vel ante vestibulum, egestas neque at, mattis nisl.

# Une liste non ordonnée

- ⊖ Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- ⊖ Nunc convallis urna turpis, ut accumsan enim aliquet vitae.
- ⊖ In vel ex fringilla, porttitor turpis vel, suscipit dui.
- ⊖ Nunc ante quam, ullamcorper ut hendrerit eu, luctus eu turpis.
- ⊖ Nam pretium tempor libero a tempor.

Braumx.net

- 1 ) Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- 2 ) Nunc convallis urna turpis, ut accumsan enim aliquet vitae.
- 3 ) In vel ex fringilla, porttitor turpis vel, suscipit dui.
- 4 ) Nunc ante quam, ullamcorper ut hendrerit eu, luctus eu turpis.
- 5 ) Nam pretium tempor libero a tempor.

# Un peu de code en java

```
package com.app

class MyClass {

    private String someStuff;
    private String anotherStuff;

    public void doStuffThings() {
        System.println("Print " + someStuff + " and " + anotherStuff);
    }

}
```

# Un peu de code en javascript

```
function setGameOver() {  
    guessField.disabled = true;  
    guessSubmit.disabled = true;  
    resetButton = document.createElement('button');  
    resetButton.textContent = 'Start new game';  
    document.body.appendChild(resetButton);  
    resetButton.addEventListener('click', resetGame);  
}
```

# Un peu de code en HTML

```
<label for="guessField">Enter a guess: </label><input type="text" id="guessField" class="guessField">  
<input type="submit" value="Submit guess" class="guessSubmit">
```

# Un peu de code en Go

```
package main

import "fmt"

func main() {

    if 7%2 == 0 {
        fmt.Println("7 is even")
    } else {
        fmt.Println("7 is odd")
    }

    if 8%4 == 0 {
        fmt.Println("8 is divisible by 4")
    }

    if num := 9; num < 0 {
        fmt.Println(num, "is negative")
    } else if num < 10 {
        fmt.Println(num, "has 1 digit")
    } else {
        fmt.Println(num, "has multiple digits")
    }
}
```



# Un peu de code en Rust

```
use std::convert::TryFrom;
use std::convert::TryInto;

#[derive(Debug, PartialEq)]
struct EvenNumber(i32);

impl TryFrom<i32> for EvenNumber {
    type Error = ();

    fn try_from(value: i32) -> Result<Self, Self::Error> {
        if value % 2 == 0 {
            Ok(EvenNumber(value))
        } else {
            Err(())
        }
    }
}

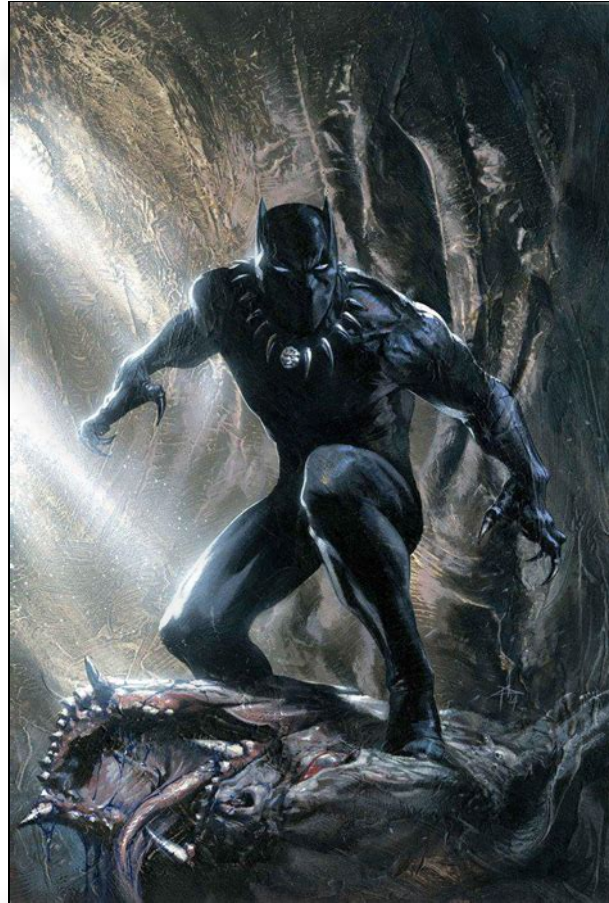
fn main() {
    // TryFrom

    assert_eq!(EvenNumber::try_from(8), Ok(EvenNumber(8)));
    assert_eq!(EvenNumber::try_from(5), Err(()));

    // TryInto

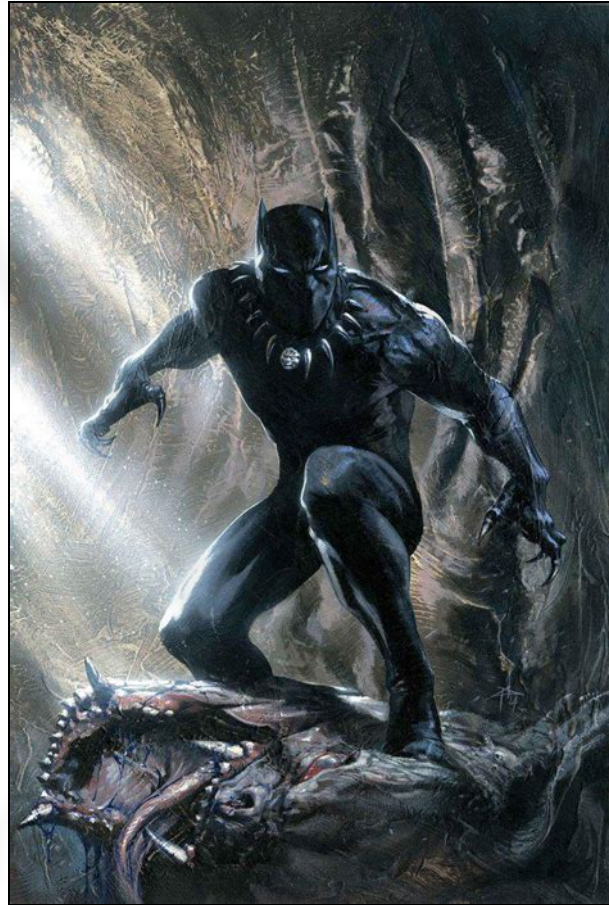
    let result: Result<EvenNumber, ()> = 8i32.try_into();
    assert_eq!(result, Ok(EvenNumber(8)));
    let result: Result<EvenNumber, ()> = 5i32.try_into();
    assert_eq!(result, Err(()));
}
```

# Une image



# Une image avec du texte

L'idéal ici serait de pouvoir faire un habillage de l'image, mais la propriété *CSS float* ne semble pas fonctionner ici.



# Une vidéo



"Internet est source de connaissances quasiment illimitée.  
Mais, par pitié, vérifiez vos sources avant de poster !"  
Napoléon Bonaparte

Bauxnet