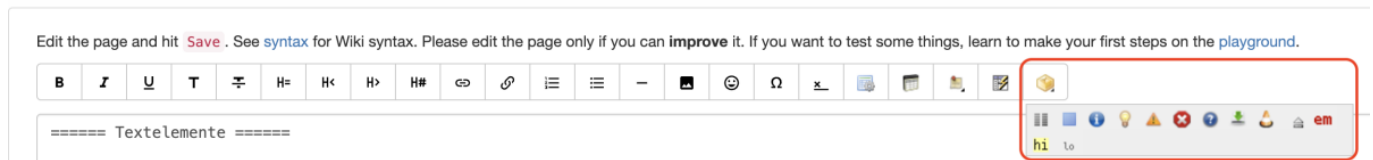


Textelemente

Wir verwenden für Infoboxen etc. Elemente aus dem WRAP-Plugin. Dieses ist über die Toolbar zugänglich:



Beispiele

Einfache Box

```
<WRAP box round center 60%></WRAP>
```



Info

```
<WRAP info round center 60%></WRAP>
```



Tip

```
<WRAP tip round center 60%></WRAP>
```



Important

```
<WRAP important round center 60%></WRAP>
```



Alert

```
<WRAP alert round center 60%></WRAP>
```



**Help**

```
<WRAP help round center 60%></WRAP>
```

**Download**

```
<WRAP download></WRAP>
```

**Todo**

```
<WRAP todo round center 60%></WRAP>
```

Ohne Round**Einfache Box**

```
<WRAP box center 60%></WRAP>
```

Andere position: left, right, center**Einfache Box**

```
<WRAP box round left  
40%></WRAP>
```

Einfache Box

```
<WRAP box round right  
40%></WRAP>
```

Oder auch Inline:

Einfach mit wrap anstat WRAP `info`, `help`, `alert`, `important`, `tip`, `download`, `todo` and `round box` and `danger`, `warning`, `caution`, `notice`, `safety`.

```
<wrap info>info</wrap>, <wrap help>help</wrap>, ...
```

Code

Der Code-Tag kann mit der Sprache ergänzt werden um Syntax highlighting zu aktivieren:

`<code java> ... </code>`

```
public class Fahrzeug {  
  
    private String bezeichnung;  
    private double geschwindigkeit;  
  
    public Fahrzeug(String bezeichnung, double geschwindigkeit){  
        this.bezeichnung = bezeichnung;  
        this.geschwindigkeit = geschwindigkeit;  
    }  
  
    public double getFahrtdauer(int fahrtstrecke){  
        double zeit = (fahrtstrecke / geschwindigkeit) * 60;  
        return zeit;  
    }  
}
```

Um Code etwas besser vom Fliesstext abzuheben, kann er sehr gut in WRAP-Elemente gepackt werden:

```
public class Fahrzeug {  
  
    private String bezeichnung;  
    private double geschwindigkeit;  
  
    public Fahrzeug(String bezeichnung, double  
geschwindigkeit){  
        this.bezeichnung = bezeichnung;  
        this.geschwindigkeit = geschwindigkeit;  
    }  
  
    public double getFahrtdauer(int fahrtstrecke){  
        double zeit = (fahrtstrecke / geschwindigkeit) * 60;  
        return zeit;  
    }  
}
```

Code als Files

Möchte man Code direkt als herunterladbares File zur Verfügung stellen, so bietet sich der `<file>`-Tag an.

`<file java Fahrzeug.java>...</file>`

Fahrzeug.java

```
public class Fahrzeug {  
  
    private String bezeichnung;  
    private double geschwindigkeit;  
  
    public Fahrzeug(String bezeichnung, double geschwindigkeit){  
        this.bezeichnung = bezeichnung;  
        this.geschwindigkeit = geschwindigkeit;  
    }  
  
    public double getFahrtdauer(int fahrtstrecke){  
        double zeit = (fahrtstrecke / geschwindigkeit) * 60;  
        return zeit;  
    }  
}
```

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/howto/wiki/textelemente>

Last update: **2024/03/28 14:07**

