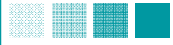


# Flex Styles

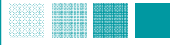
Roman Bartoli



# Gliederung

- Was sind Styles?
- Styles setzen
- Style Definition
- Selektoren
- Styles in Actionscript





# Was sind Styles

- Über Styles lässt sich das visuelle Erscheinungsbild von Komponenten verändern
- Beispiel:
  - Farbe
  - Transparenz
  - Schrift
  - Rahmen
  - Abstände





## Was sind Styles

- Über Styles lassen sich nur Attribute verändern
- Vorhandenes Design Theme gibt Grundstruktur vor
- Komplexere Formen müssen über *Skinning* angepasst werden

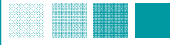


Styling



Skinning



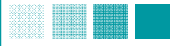


## Styles setzen

- Es gibt eine klare Hierarchie
- Jede Stufe überschreibt die darunter liegende

1. Instance/ Inline Style
2. Class Style
3. Type Style
4. Global Style





## Instance/ Inline Style

- Bezieht sich auf eine Instanz des MXML Objekts
- Bläht den Quelltext unnötig auf
- Sollte bei komplexen Projekten wenn möglich vermieden werden

Beispiel:

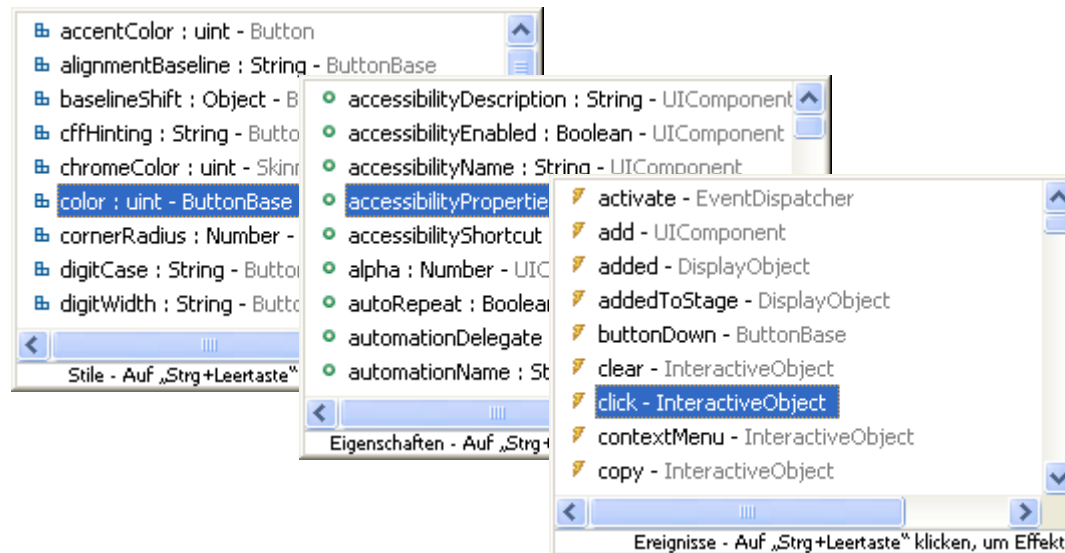
```
<s:Label color="#FF0000" fontsize="18" />
```



## Instance/ Inline Style

- Wie erkennt man die Styles in MXML?
- Mehrmaliges Betätigen von Steuerung + Leertaste gliedert die Eigenschaften in Events, Effects, Styles

- **<s:Button**



# Class Style

- Verschiedene Styles werden zu einer Klasse zusammengefasst
- Styles werden in CSS-Blöcken definiert
- Flex hat das CSS-Konzept übernommen

Beispiel:

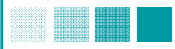
```
.classStyleExample {  
    color: #ff0000;  
    font-size: 18;  
}
```



```
<s:Label styleName="classStyleExample" />
```







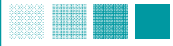
## Type Style

- Gelten für einen bestimmten Komponententyp
- Wird automatisch auf alle Komponenten des Typs (samt Unterklassen) angewendet

Beispiel:

```
@namespace s "library://ns.adobe.com/flex/spark";  
@namespace mx "library://ns.adobe.com/flex/mx";  
  
s|Button{  
    color: #00ff00;  
}
```





# Global Style

- Gelten für alle Komponenten
- Sinnvoll um global alle Defaultwerte zu überschreiben

Beispiel:

```
global{  
    font-size: 18;  
}
```





## Style Definition

- Alle CSS-Styles können entweder in einer externen CSS-Datei oder in MXML definiert werden
- Bei komplexen Projekten sollte der CSS-Datei der Vorzug gegeben werden

### In MXML:

```
<fx:Style>
    .classStyleExample {
        color: #ff0000;
        font-size: 18;
    }
</fx:Style>
```

### Aus externer CSS-Datei:

```
<fx:Style source="styles.css">
```



# Selektoren

- ID-Selektor

```
#loginPanel {  
    color: #ff0000;  
}
```

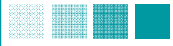
- Descendant-Selektor

```
s|Panel s|Button {  
    color: #ff0000;  
}
```

- Pseudo-Selektor

```
s|Button:up {  
    color: #ff0000;  
}  
s|Button:over {  
    color: #00ff00;  
}  
s|Button:down {  
    color: #0000ff;  
}
```





# Styles in Actionscript

- Alle Styles können auch zur Laufzeit in AS geändert werden

Beispiel:

```
private function styleButton():void {  
    var styleDeclaration:CSSStyleDeclaration = new  
        CSSStyleDeclaration();  
    styleDeclaration.setStyle("fontSize", 0xFF0000);  
    styleDeclaration.setStyle("color", "20");  
    styleManager.setStyleDeclaration(".myStyle",  
        styleDeclaration, true);  
    button.styleName="myStyle";  
}  
  
private function styleButton2():void {  
    button.setStyle("fontSize", 0xFF0000);  
    button.setStyle("color", "20");  
}
```

