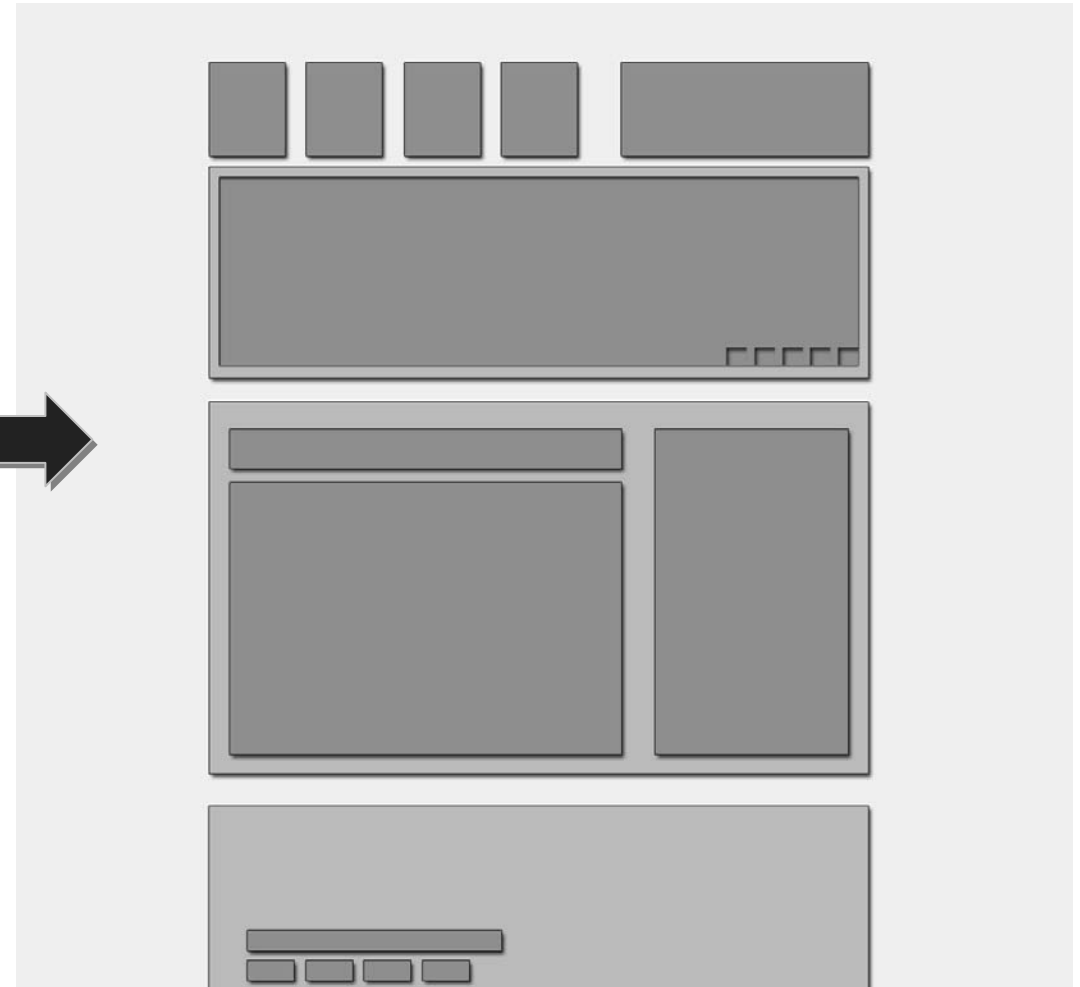


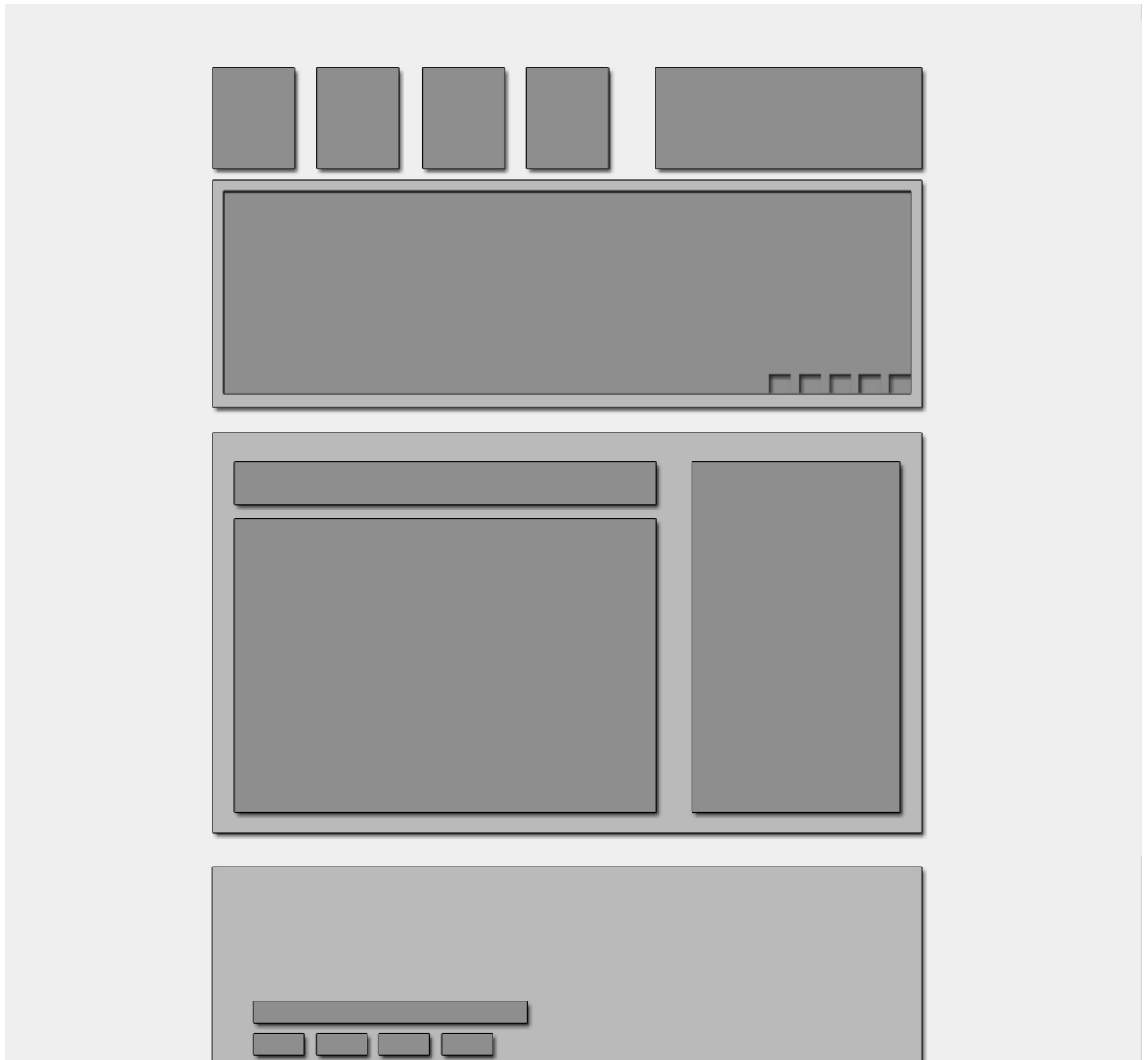
## IT-Zertifikat: Allgemeine Informationstechnologien II – Übungsblatt jQuery

Für unser Miniredaktionssystem, das wir im Laufe der nächsten Wochen / Monate entwickeln werden, verwenden wir eine der Website <http://www.artisanmedia.com/> ähnliche Aufteilung der Website:



## Aufgabe 1 – Der CSS-Part I

Vergeben Sie Namen für die einzelnen Elemente der Website und überlegen Sie, wie die unterschiedlichen Seitenbereiche in CSS und HTML repräsentiert werden könnten (i.e. welche Seitenbereiche kommen einmal vor, welche Seitenbereiche mehrfach?).



## Aufgabe 2 – Der CSS-Part II

Wir beschäftigen uns zunächst mit dem oberen Bereich des Layouts:



Erstellen Sie eine HTML-Website mit entsprechendem Cascading Style Sheet, um die Seitenbereiche anzulegen:

- Die Website ist umschlossen von einem Div-Container `#dose`:  
`#dose { margin-left: auto; margin-right: auto; width: 750px; padding: 5px; }`
- Die obere Navigationsleiste `#topnavigation` hat eine Breite von 425px, eine Höhe von 105px und eine beliebige Hintergrundfarbe, z.B. `#cccccc`. Die obere Navigationsleiste enthält vier Div-Container, die jeweils mit einer eigenen ID ausgestattet werden: `#topbutton1`, `#topbutton2`, etc. Jeder `#topbutton` soll um eine Hintergrundfarbe ergänzt werden.
- Neben den vier Navigationselementen befindet sich das Logo `#logo` der Website. Das Logo verfügt über eine Höhe von 105px, eine Breite von 275px und eine Hintergrundfarbe.
- Jeder `#topbutton` ist 85px hoch und 75px breit, verfügt über eine Hintergrundfarbe, einen Rand (margin) von 10px. Der rechte Rand ist zudem auf 15px gesetzt: `margin-right: 15px`.
- Um die Elemente anzuordnen:
  - `#topnavigation` wird rechts umflossen (`float: left`) vom `#logo` (`float: right`).
  - Jeder `#topbutton` wird rechts von seinem Nachbarn umflossen: `float: left`.
- Unterhalb der Navigationsleiste befindet sich eine Teaserbox `#teaserbox`. Die Höhe der Teaserbox beträgt 275px, die Breite beträgt 740px. Der Abstand (padding) vom äußeren Rand der Teaserbox zu Elementen der `#teaserbox` beträgt 5px. Der Abstand der Teaserbox zum oberen Navigationsbereich beträgt 125px: `margin-top: 125px`. Die `#teaserbox` ist durch eine Hintergrundfarbe gekennzeichnet.
- Innerhalb der `#teaserbox` befindet sich der Div-Container `#currentimage` mit einer Höhe und Breite von 100% und einer Hintergrundfarbe.
- Innerhalb der `#teaserbox` befinden sich fünf `#teaserbutton` (`#teaserbutton1`, `#teaserbutton2`, etc.) mit einer Höhe und Breite von 25px, einer Hintergrundfarbe.
- Die `#teaserbutton` werden zueinander ausgerichtet über `float: right`.
- Der Abstand (margin) jedes `#teaserbutton` beträgt 3; der rechte Abstand jedes `#teaserbutton` wird anschließend auf 0 gesetzt und der Abstand nach oben ist ein negativer Abstand von -25px:  
`margin: 3px;`  
`margin-right: 0;`  
`margin-top: -25px;`

## Aufgabe 3 – Der Interaktionspart: jQuery & Co.

### 3.1 Elemente mit Inhalt füllen

Die Navigationselemente #topbutton1 bis #topbutton4 sollen per jQuery die folgenden Beschriftungen erhalten:

- #topbutton1: Home
- #topbutton2: Aktuelles
- #topbutton3: Login
- #topbutton4: Kontakt

### 3.2 Das Verhalten von DOM-Elementen beeinflussen

Die Hintergrundfarbe der Navigationselemente #topbutton1 bis #topbutton4 soll sich verändern, wenn sich der Mousezeiger über dem entsprechenden Element befindet. Hierfür ergänzen wir das Stylesheet um die Klasse .mouseovertopbutton:

```
.mouseovertopbutton
{
    background-color: #000000 !important;
    color: #ffffff;
}
```

Der jQuery-Code zur Behandlung des „Hover“-Ereignisses schaut wie folgend aus:

```
jQuery('#topbutton1').hover(function(){
    // Mouse befindet sich über #topbutton1: CSS-Klasse .mouseovertopbutton hinzufügen
},
function()
{
    // Mousezeiger verlässt #topbutton1: CSS-Klasse .mouseovertopbutton per jQuery entfernen
});
```

Ergänzen Sie den jQuery-Code, so dass sich die Hintergrundfarbe jedes #topbutton beim Herüberfahren mit der Mouse verändert.

### 3.3 Das Verhalten von DOM-Elementen beeinflussen II

Erstellen Sie eine weitere CSS-Klasse, um festhalten zu können, welche Elemente der #topnavigation angeklickt wurden:

```
.activatedtopbutton
{
    background-color: red !important;
    color: #ffffff;
}
```

Fügen Sie jQuery-Code hinzu, um die Klasse des Elementes beim Klick mit der linken Mausekante auf „activatedtopbutton zu ändern:

```
jQuery("#topbutton1").click(function (){

    // Klasse des angeklickten Elementes ändern

});
```

### 3.4 Inhalte einfügen

Bei Aufruf der Website soll eine Grafik in den div-Container #currentimage geladen werden. Verwenden Sie hierzu die jQuery-Funktion appendTo ( ).

### 3.5 Inhalte ersetzen

Statten Sie jeden #teaserbutton (#teaserbutton1, #teaserbutton2, etc.) mit click-Funktionalität aus: Bei einem Klick mit der linken Mausekante auf einen #teaserbutton wird das aktuelle Bild des div-Containers #currentimage ersetzt durch ein anderes Bild. Verwenden Sie zum Ersetzen die jQuery-Funktion replaceWith( ).

### 3.6 Inhalte ausblenden

Fügen Sie Ihrem HTML-Dokument den Div-Container #maincontent hinzu. #maincontent befindet sich als letztes Element innerhalb des Hauptcontainers #dose. #maincontent hat die folgenden Eigenschaften:

```
#maincontent
{
    height: 500px;
    width: 100%;
    background-color: #dddddd;
    margin-top: 25px;
}
```

Füllen Sie #maincontent mit Beispieldaten, z.B. von <http://www.loremipsum.de/>.

Legen Sie zunächst per jQuery fest, dass #maincontent nach dem Aufruf der Website nicht sichtbar ist. Über einen Klick mit der linken Mausekante auf den Button „Aktuelles“ wird #maincontent situationsabhängig ein- oder ausgeblendet: Wird auf „Aktuelles“ geklickt, so wird #maincontent eingeblendet (fadeIn) oder ausgefahren (slideDown), wird erneut geklickt, so wird #maincontent ausgeblendet (fadeOut) oder eingefahren (slideUp). Verwenden Sie hierzu eine Konstruktion analog des auf dem Handout „jQuery“ wiedergegebenen Codes unter „Effekte“ (S. 4).