3. Zeichen und Schrift

Weitere Informationen: http://de.selfhtml.org/ oder http://w3schools.org

Nachtrag:
Zielgenaue Verweise: Dokumentinterne Anker

- Hinter jeder Verweisadresse kann (mit # abgetrennt) eine Stelle in dem adressierten Dokument spezifiziert werden.
  - Ansprechen des Zielankers:

```
<a href="xyz.html#hierher">Text</a>
```

- Alte Methode zur Deklaration des Zielankers (z.B. in xyz.html):

```
<a name="hierher">Text</a>
```

- HTML5-Methode zur Deklaration des Zielankers (z.B. in xyz.html):

```
<p id="hierher">Text</p>
```
(falls Element vorhanden) oder
```
<span id="hierher">Text</span>
```

links.html

Tabellen (1)

- Aufteilen der Fläche in Zeilen und Spalten in flexibler Weise
  - Klassische Tabellen, Matrizen
  - Allgemeines Hilfsmittel zum Layout (bei unsichtbar gemachten Trennlinien)
  - Achtung: Tabellen werden meist erst nach vollständigem Laden angezeigt
- Allgemeine Tabellenform:

```
<table>
   <tr>  <th></th> <th></th> <th></th>  </tr>
   <tr>  <td></td> <td></td> <td></td>  </tr>
   <tr>  <td></td> <td></td> <td></td>  </tr>
</table>
```

Mit **<thead>**, **<tbody>** und **<tfoot>** kann man logische Bereiche definieren.

Tabellen (2)

- Vordefinition der Spaltenbreite (schnellere Anzeige!)
  - **`<colgroup> <col width=...> ... </colgroup>`**
- Unregelmässige Zellen einer Tabelle
  - Zelle über mehrere Spalten: Attribut **`colspan="`*`n`*`"`** in **`<th>`** und **`<td>`**
  - Zelle über mehrere Zeilen: Attribut **`rowspan="`*`n`*`"`** in **`<th>`** und **`<td>`**
- Rahmen
  - Veraltet: Attribut **`border="`*`n`*`"`** in **`<table>`**
  - Besser mit Cascading Style Sheets, siehe gleich...
- Abstände, Textformatierung, Ausrichtung etc.
  - mit Cascading Style Sheets, siehe gleich...

table.html

3. Zeichen und Schrift

3.1 Medien Zeichen, Text, Schrift

3.2 Mikro-Typografie: Zeichensätze

3.3 Makro-Typografie: Gestalten mit Schrift

3.4 Hypertext und HTML 

- Allgemeines
- Textstrukturierung
- Tabellen
- Cascading Style Sheets 
- Strukturierte Seiten
- Medieneinbettung

Weitere Informationen: http://de.selfhtml.org/ oder http://w3schools.org

Literatur:

David Sawyer McFarland: CSS3 - The missing manual , 3rd ed., O'Reilly 2013

Cascading Style Sheets (CSS)

- Von HTML prinzipiell unabhängige Sprache zur Beschreibung von Formatierungsinformation
  - Standardisierung durch W3C
  - Besonders für HTML geeignet
- Entstehungsgeschichte:
  - Vielzahl von "Standard-Attributen" in vielen HTML-Elementen (align, pos, color, font, ...)
  - Vereinheitlichung in CSS (aktuelle Version 2.1, CSS 3 in Entwicklung)
- Ablösung "alter" Konstrukte zugunsten CSS-beschriebener Styles: Empfehlung in HTML bis Version 4, verpflichtend ab HMTL5
  - Alte Schreibweise (nicht mehr benutzen):
    ```
    <p><font size="7">Text</font></p>
    ```
  - Schreibweise mit CSS-Syntax, Universalattribut **`style`** (nicht empfohlen):
    ```
    <p style="font-size:7">Text</p>
    ```
  - Empfehlenswert für modernes HTML: Rein logische Auszeichnung
    - » Formatierung separat festgelegt (in CSS Style Sheet)

CSS-Syntax: Deklarationen

- *Deklaration:* Eigenschaft-Wert-Paar

  `Eigenschaft : Wert` z.B. **`font-style:italic;`**

  – Wenn als Wert eines HTML-Attributs: Anführungszeichen "" empfehlenswert

  – Mit Strichpunkt beenden (mehrere Deklarationen in Folge)

  z.B. **`font-style:italic; font-size:large;`**

- Anführungszeichen für Werte (z.B. bei Leerzeichen im Wert)

  – Einfache Anführungszeichen `''`

  z.B. **`font-family:'Times New Roman';`**

- Mehrere Werte (Argumente) für eine Eigenschaft

  – Abtrennen mit Leerzeichen

  z.B. **`border:3px solid black;`**

- Mehrere Alternativwerte (Sequenz) für eine Eigenschaft

  – Abtrennen mit Komma

  z.B. **`font-family:'Times New Roman', 'Times', serif;`**

CSS-Eigenschaften, Beispiel Schriftformatierung

- Eigenschaften zur Schriftformatierung:
  - **`font`** Zusammenfassung anderer Eigenschaften
  - **`font-family`** Gewünschte Schrift(en) mit Priorisierung
  - **`font-style`** Kursiv / normal
  - **`font-variant`** Kapitälchen *(small caps)* / normal
  - **`font-size`** Größe (numerisch oder ungenau)
  - **`font-weight`** Strichstärke (fett / mager)
  - **`font-stretch`** Laufweite
  - **`word-spacing`** Wortabstand
  - **`letter-spacing`** Zeichenabstand
  - **`color`** Farbe
  - …

Weitere CSS-Eigenschaften

- Schriftformatierung (auch mit Schriftartendatei)
- Ausrichtung und Absatzkontrolle
- Außenrand und Abstand
- Innenabstand
- Rahmen
- Hintergrundfarben und -bilder
- Listenformatierung
- Tabellenformatierung
- Positionierung und Anzeige von Elementen
- Layouts für Printmedien
- Sound-Kontrolle für Sprachausgabe
- Anzeigefenster

Einbindung von CSS in HTML

- *Inline styles:*
  - **`style`**-Attribut bei HTML-Tags benutzen
- *Internal style sheet:*
  - Ablage von zentralen Stildefinitionen im Kopfbereich der HTML-Datei

```
<style type="text/css">
... Stildefinitionen ...
</style>
```

  - Wegen Problemen älterer Browser oft Stildefinitionen als Kommentar
- *External style sheet:*
  - Ablage von zentralen Stildefinitionen in separater CSS-Datei (.css)
  - Enthält nur Stildefinitionen, kein HTML
  - Einbindung in HTML-Datei:

```
<link rel="stylesheet" type="text/css"
href=Dateireferenz>
```

CSS-Syntax: Stildefinition

- *Stildefinition* besteht aus *Selektor* und *Deklarationsblock:*
  `Selektor { Deklaration* }`
- Beispiel:
```
h1 { font-size: 48pt;
     font-style: italic;
     border-bottom: solid thin black; }
```
- Mögliche Selektoren:
  - *Element-Selektor:* Alle Elemente eines Elementtyps (z.B. **`p`**)
  - *Klassen-Selektor:* Selbstdefinierte Stilklasse, frei gewählter Name
    - » Selektor beginnt mit Punkt, Anwendung mit HTML **`class`**-Attribut
  - *ID-Selektor:* Einzelnes eindeutiges Element, frei gewählter Name
    - » Selektor beginnt mit **`#`**, Anwendung mit HTML **`id`**-Attribut
  - *Attributselektor:* Elemente mit gegebenem Attributwert, ***`[attr=val]`***
  - *Gruppenselektor* (mehrere Elemente, gleicher Stil): mit Komma trennen
  - *Universalselektor* ***`*`****:* alle Elemente
  - *Pseudo-Klassen:* Beginnen mit Doppelpunkt (z.B. **`:visited`**)
  - *Kombinationsselektoren:* Nachfahr, Kind, Geschwister

Beispiel zu CSS (Internes Stylesheet)

```
<!DOCTYPE html>
<html>
  <head>
     <meta charset="UTF-8">
     <title>Beispiel zu CSS</title>
     <style>
           p     {font-family:Times; font-size:20pt}
           h1    {font-family:Verdana; color:red}
     </style>
  </head>
  <body>
     <h1>&Uuml;berschrift 1</h1>
     <p>Absatz 1</p>
     <h1>&Uuml;berschrift 2</h1>
     <p>Absatz 2</p>
     <h1>&Uuml;berschrift 3</h1>
     <p>Absatz 3</p>
  </body>
</html>
```

styles.html

Beispiel zu CSS (Externes Stylesheet)

```
<!DOCTYPE html>

<html>
  <head>
     <meta charset="UTF-8">
     <title>Beispiel zu CSS</title>
     <link rel="stylesheet" type="text/css" href="styles.css">
  </head>
  <body>
     <h1>&Uuml;berschrift 1</h1>
     <p>Absatz 1</p>
     <h1>&Uuml;berschrift 2</h1>
     <p>Absatz 2</p>
     <h1>&Uuml;berschrift 3</h1>
     <p>Absatz 3</p>
  </body>
</html>
```

stylesfile.html

Datei **styles.css** (im gleichen Verzeichnis):

```
p     {font-family:Verdana; font-size:16pt}
h1    {font-family:Verdana; color:green}
```

Beispiel zu CSS-Klassen

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Beispiel zu CSS-Klassen</title>
    <style>
      *        {font-family: sans-serif;}
      .kursiv  {font-style: italic;}
      #start   {font-style: italic; font-weight: bold;
      p.rot    {color: red;}
    </style>
  </head>
  <body>
    <h1>Überschrift mit
      <span class="kursiv">besonderem Stichwort</span></h1>
    <p id="start">Erster Absatz</p>
    <p class="rot">Zweiter Absatz</p>
    <p class="kursiv">Dritter Absatz</p>
    <p>Vierter Absatz</p>
    <p class="kursiv">Fünfter Absatz</p>
    <blockquote class="rot">Zitat</blockquote>
  </body>
</html>
```

styleclasses.html

Anmerkungen zu CSS-Klassen

- *Inline-Element* **`<span> ... </span>`**
  - Für Text und Zeilenelemente, nur zur Formatierung
- *Allgemeines Blockelement* **`<div> ... </div>`**
  - Kann beliebige Blockelemente enthalten, z.B. auch Grafiken
  - Anwendung von Formatierung, incl. Positionierung, Sichtbarkeit
- Einschränkungen der Gültigkeit von Stildefinitionen:
  - Klassen auf bestimmte Elemente beschränken
    - » z.B. **`.rot`**
  - Stil nur anwenden, wenn Element unterhalb eines bestimmten anderen
    - » z.B. **`tbody tr {...}`** oder **`#haupttext p {…}`**
- Vererbung:
  - Viele Eigenschaften vererben sich auf untergeordnete Elemente
    - » nicht alle (siehe z.B. Rahmen)!
  - Konflikte (auch mit Browser-Einstellungen):
    Bei Konflikten gewinnt die direktere Definition bzw. das spezifischere Format

3. Zeichen und Schrift

3.1 Medien Zeichen, Text, Schrift

3.2 Mikro-Typografie: Zeichensätze

3.3 Makro-Typografie: Gestalten mit Schrift

3.4 Hypertext und HTML 

- Allgemeines
- Textstrukturierung
- Cascading Style Sheets
- Tabellen
- Strukturierte Seiten 
- Medieneinbettung

Weitere Informationen: http://de.selfhtml.org/ oder http://w3schools.org

Literatur:

Peter Kröner: HTML5, Open Source Press 2010

Strukturierte Seiten (Layout)

- Moderne Webseiten haben ein klar definiertes Layout (Satzspiegel)
  - Bestandteile mit verschiedener Funktion
    (z.B. Kopf, Navigation, Hauptteil, Fußzeile)
  - Freies 2-dimensionales Layout
    (oft nebeneinander platzierte Einheiten)
- Realisierungsmöglichkeiten:
  - Framesets (Element <frame>): Veraltet und nicht empfohlen
  - Tabellen:
    - » (Immer noch) verbreitet und recht effektiv
    - » Aber: Keine Trennung von logischer Struktur und Layout
  - Blockelemente (Element):
    - » Elegante moderne Lösung zusammen mit CSS
    - » Dominanz des -Elements ("Divitis")
  - HTML5-Strukturelemente und CSS:
    - » Derzeit beste Entkopplung von Struktur und Layout

Definition der logischen Struktur

- Welche separaten Bereiche enthält unsere Seite?
  - Möglichst übergreifend über alle Seiten eines Webauftritts
- Beispiel:
  - Kopfbereich:
    - » Für alle Seiten gleich
  - Navigationsbereich
    - » Für alle Seiten gleich; enthält Liste von Einträgen (Links)
  - Hauptteil:
    - » Soll Liste von Einträgen (Artikeln) enthalten
  - Fußzeile:
    - » Für alle Seiten gleich; kurzer Text (Impressum)

Logische Struktur in HTML 4

```
<body>
      <div id="header">
            <h1>Structured Page (HTML4)</h1>
      </div>
      <div id="nav">
            <ul>
                  <li><a href=#">Home</a></li>
                  <li><a href=#">Contact</a></li>
            </ul>
      </div>
      <div id="main">
            <h1>This is the main content area of the page.</h1>
            <p>This is the first content article.</p>
            <p>This is the second content article.</p>
      </div>
      <div id="footer">
            <p>Heinrich Hußmann, LMU, 2010</p>
      </div>
</body>
```

Logische Struktur in HTML5

```
<body>
   <header>
      <h1>Structured Page (HTML5)</h1>
   </header>
   <nav>
      <ul>
         <li><a href="#">Home</a></li>
         <li><a href="#">Contact</a></li>
      </ul>
   </nav>
   <section id="main">
      <h1>This is the main content area of the page.</h1>
         <article>
            <p>This is the first content article.</p>
         </article>
         <article>
            <p>This is the second content article.</p>
         </article>
   </section>
   <footer>
      <p>Heinrich Hu&szlig;mann, LMU, 2010</p>
   </footer>
</body>
```

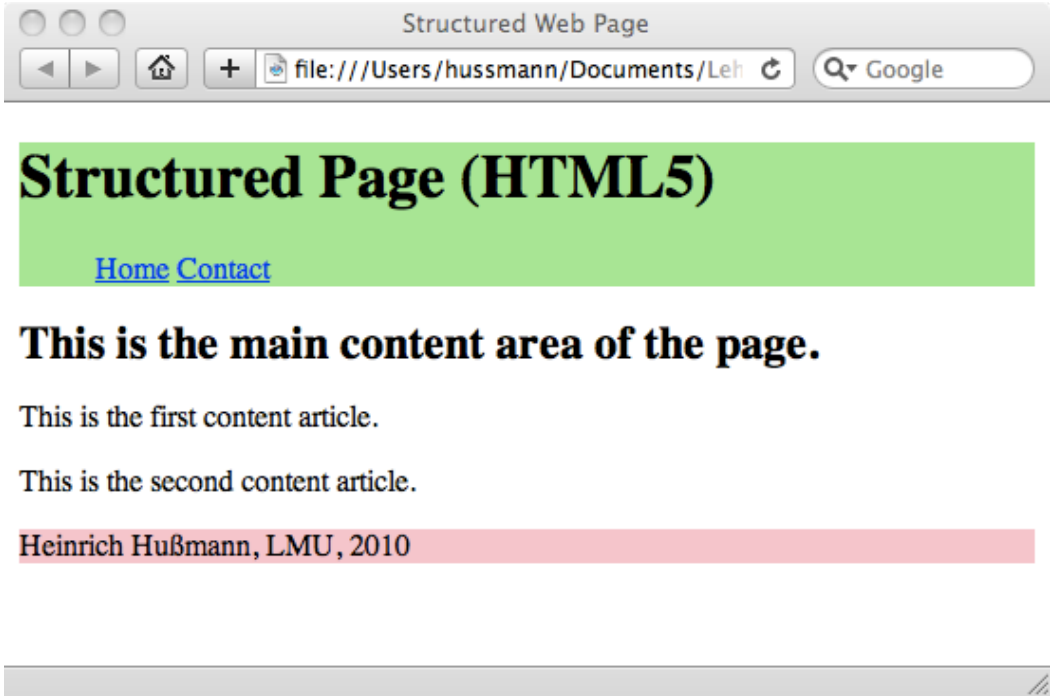
Layouts

- Statisches Layout:
  - Geht meist von festem Anzeigebereich aus (Höhe x Breite in Pixel)
  - Verkleinerung schneidet ab, Vergrößerung schafft Leerraum
  - Derzeit dominant
- Flüssiges Layout (“liquid”):
  - Behält Standard-Schema des Layouts bei
  - Anpassung an aktuelle Fenstergröße, z.B. durch prozentuale und relative Angaben im Layout
- Responsive Web Design (anpassungsfähiges Design)
  - Begriff von Ethan Marcotte (“A List Apart”, 2010)
  - Layout passt sich den Möglichkeiten des Ausgabegeräts an
    - » z.B. “Media Queries” und Laden eines passenden Style Sheets

Einfaches Vertikal-Layout mit CSS

- CSS-Stylesheet zum Beispiel:

```
<style>
      header {
           background-color:lightgreen;
      }
      footer {
           background-color:pink;
      }
      nav li {
           display:inline;
      }
      header h1 {
           font-size:2em;
      }
      #main h1 {
           font-size:1.5em;
      }
</style>
```

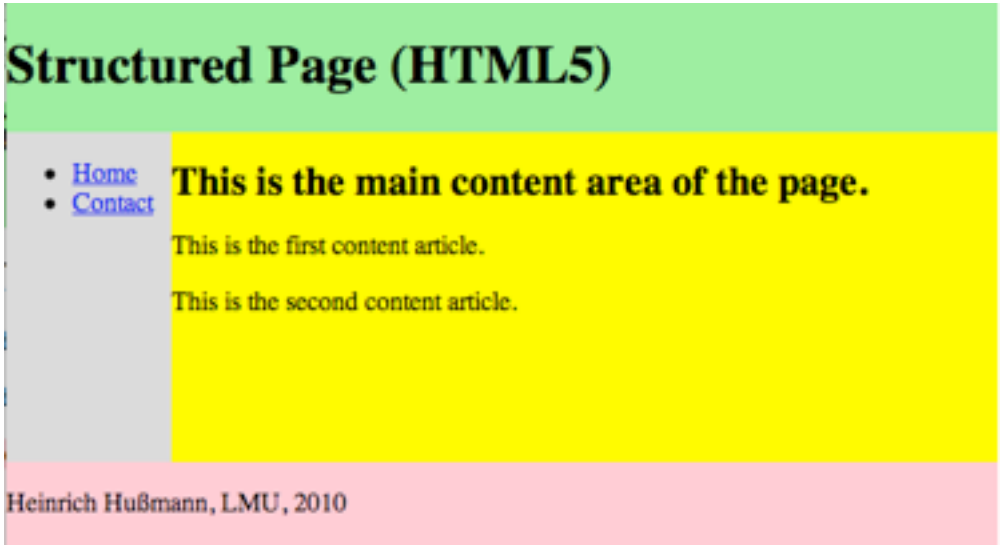


html5struct_vert.html

Mehrspalten-Layout mit CSS (feste Größen)

```
header {
    background-color:lightgreen;
    height:80px;
    width:600px;
    position:absolute;
    left:0px;
    top:0px;
}
nav {
    height:200px;
    width:100px;
    background-color:lightgrey;
    position:absolute;
    top:80px;
    left:0px;
}
#main {
    background-color:yellow;
    position:absolute;
    height:200px;
    width:500px;
    left:100px;
    top:80px;
}
```

```
footer {
    background-color:pink;
    width:600px;
    position:absolute;
    top:280px;
    left:0px;
}
header h1 {
    font-size:2em;
}
#main h1 {
    font-size:1.5em;
}
```



html5struct_fixed.html

Mehrspalten-Layout mit CSS ("flüssig")

```
header {
    background-color:lightgreen;
    height:20%;
    width:100%;
    position:absolute;
    left:0%;
    top:0%;
}
nav {
    height:70%;
    width:10%;
    background-color:lightgrey;
    position:absolute;
    top:20%;
    left:0%;
}
#main {
    background-color:yellow;
    position:absolute;
    height:70%;
    width:90%;
    left:10%;
    top:20%;
}
```

```
footer {
    background-color:pink;
    height:10%;
    width:100%;
    position:absolute;
    top:90%;
    left:10%;
}...
```

html5struct_liquid.html

Entwicklung von CSS

- Dynamische Änderungen der Anzeige:
  - Z.B. Markierung von Ankern an Mausposition
  - Prozedurale (Skript-basierte) Lösung -> Deklarative Lösung in CSS3
- Unterstützung von Animationen
  - z.B. bei Erscheinen/Entfernen von angezeigten Elementen
- Vom druck-formatiertem Text zu graphischen Multimedia-Dokumenten
  - Z.B. Abgerundete Begrenzung von Bereichen
- CSS3 ist in Entwicklung, und es wird weiter gehen...

Runde Boxen und Transitionen in CSS3

```
p  {
  background-color:red;
  transition-property:background-color;
  transition-duration:2s;
  ...
  box-shadow: 5px 5px 3px lightgrey;
}
p:hover   {
  background-color:black;
}
```

```
<body>
  <p><a href="#">Click here!</a></p>
</body>
```

css3anim.html

3. Zeichen und Schrift

3.1 Medien Zeichen, Text, Schrift

3.2 Mikro-Typografie: Zeichensätze

3.3 Makro-Typografie: Gestalten mit Schrift

3.4 Hypertext und HTML 

- Allgemeines
- Textstrukturierung
- Cascading Style Sheets
- Tabellen
- Framesets
- Medieneinbettung 

Weitere Informationen: http://de.selfhtml.org oder http://w3schools.org

Integration von Bildern

- Bilder einbinden mit **<img>**
- Attribut **src** gibt Quelle an (auch von anderen Servern möglich)
  - Achtung Copyright-Fragen!
- Größenangaben mit **width** und **height**
  - Bei Angabe beider Werte Verzerrung möglich
- Bilder können auch als Inhalt eines Verweises vorkommen
  - z.B. grafische Navigationsleisten

```
<html> ...
 <body>
   <h1>Ein JPEG-Bild des Eiffelturms</h1>

   <p><img src="Eiffelturm.jpg" width="350" alt="Eiffelturm">
   </p>

 </body>
</html>
```

Integration anderer Dateien

- Prinzipiell alle Dateien einbettbar
  - mit dem **<object>**-Tag (standardkonform)
  - als Hyperlinks
- Leider Behandlung von Medien in Browsern uneinheitlich!
- HTML5 versucht Vereinheitlichung zu erreichen.
  - HTML5 Audio:

```
<audio src="xyz.ogg" autoplay>
  Your browser does not support the <code>audio</code> element.
</audio>
```

  - HTML5 Video:

```
<video controls>
  Your browser does not support the <code>video</code> element.
  <source src="big_buck_bunny_480p_stereo.ogg" type="video/ogg">
  <source src="big_buck_bunny_480p_surround-fix.avi">
</video>
```

MIME

- MIME = Multipurpose Internet Mail Extensions
  - In HTML mit dem **`type`**-Attribut an vielen Stellen angebbar (z.B. **`<link>, <object>`**)
  - Erleichtert dem Browser (bzw. seinem Benutzer) die Entscheidung, wie Dateien zu behandeln sind
  - Jeder Browser führt eine Liste der akzeptierten MIME-Extensions und Regeln für die Behandlung (z.B. speichern, Programm aufrufen)
  - Liste siehe http://www.iana.org/assignments/media-types

- Syntax:

  `Medientyp / Untertyp`

  - Medientypen: text, image, video, audio, application, ...
  - Untertypen, die auf dem Server auszuführen sind, beginnen meist mit x-
  - Hersteller- *(vendor-)*spezifische Untertypen im speziellen Unterbaum "vnd."

Barrierefreiheit von Webseiten

- **Was passiert mit Menschen, die mit Einschränkungen ihrer Nutzungsmöglichkeiten leben müssen?**
  - Äquivalent zu "barriefreien" Zugängen zu Gebäuden
  - Web tendenziell besonders wichtig und interessant für diese Nutzer
- Nutzung des WWW bei eingeschränkten Wahrnehmungs- und Aktionsmöglichkeiten
  - Seh- oder Hörbehinderung, motorische Schwächen
  - Leseschwäche, Aufnahmeschwäche, Lernschwäche
- Wichtigste Richtlinie:
  - Web Accessibility Initiative (http://www.w3.org/WAI/)
  - Übernommen in vielen nationalen Regelungen, z.B. BITV in Deutschland
- Beispiele für Regeln zur Barrierefreiheit:
  - Ergänzung grafischer Information durch textuelle Beschreibung
  - Benutzbarkeit mit Tastatur (d.h. auch mit Spracheingabe)
  - Orientierung durch klare Struktur und kleine Textblöcke erleichtern
  - Hoher Kontrast zwischen Vordergrund und Hintergrund