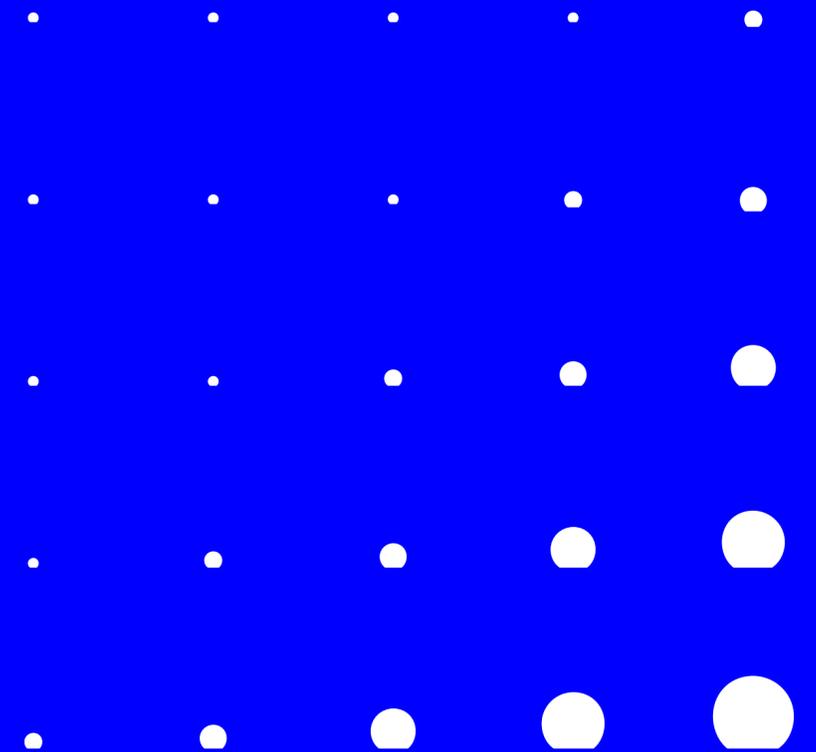


## CSS(3) erste Schritte

<https://wiki.selfhtml.org/wiki/CSS>, <https://www.edv-lehrgang.de/css-syntax/>



# Was ist CSS?

- Cascading Style Sheets (CSS) ist eine Stylesheet-Sprache für elektronische Dokumente und zusammen mit HTML eine der Kernsprachen des World Wide Webs.
- Sie ist ein lebendiger Standard und wird vom W3C beständig weiterentwickelt.
- Mit CSS werden Gestaltungsanweisungen erstellt, die zusammen mit HTML und XML (zum Beispiel bei SVG) eingesetzt werden.
- CSS wurde entworfen, um Darstellungsvorgaben von den Inhalten zu trennen.
- Wenn diese Trennung konsequent vollzogen wird, werden nur noch die inhaltliche Gliederung eines Dokumentes und die Bedeutung seiner Teile in HTML oder XML beschrieben.
  - Mit CSS werden dann die Definitionen zur Darstellung der Inhalte gesondert festgelegt, insbesondere Layout, Farben und Typografie.

# Was ist CSS?

- Vielseitige Gestaltung der physischen Darstellung
  - Schriftart, Schriftgröße, Schriftstil
  - Zeichen- und Zeilenabstand, Einrückung
  - Text- und Hintergrundfarbe
  - Rahmen
  - freie Platzierung und Überlappung von Objekten
  - unterschiedliche Formate für Druck, Bildschirm, Mobilgerät ...

# Wieso CSS?

- Strikte Trennung zwischen Inhalt und Darstellung
  - Inhalt wird logisch in HTML formatiert
  - Darstellung erfolgt separat (und parallel) in CSS
  - Arbeitsteilung zwischen Autor und Designer wird möglich
- Wiederverwendung durch Kaskadierung
- Exakte Bestimmung des Erscheinungsbilds
- Optimierung für bestimmte Ausgabemedien
  - Wichtig für barrierefreies Webdesign

Download the example [html file](#) and [css file](#)

## CSS Zen Garden

A demonstration of what can be accomplished visually through CSS-based design. Select any stylesheet from the list to load it into this page.



### Select a Design

- [Mid Century Modern](#)  
by [Andrew Lohman](#)
- [Garments](#)  
by [Dan Mall](#)
- [Steel](#)  
by [Steffen Kneller](#)
- [Apothecary](#)  
by [Trent Walton](#)

### The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible [DOMs](#), broken [CSS](#) support, and abandoned browsers. We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [W3C](#), [WaSP](#), and the major browser creators. The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

*Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.*

*We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.*

*The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.*

## THE ROAD TO ENLIGHTENMENT

Download the example [html file](#) and [css file](#)



Why participate? For recognition, inspiration, and a resource we can all refer to showing people how amazing

**ZEN**  
f CSS Design

### SO WHAT IS THIS ABOUT?

There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The HTML remains the same, the only thing that has changed is the external CSS file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. Designers and coders alike have contributed to the beauty of the web; we can always push it further.

# Vorbereitung für eine CSS-Formatierung

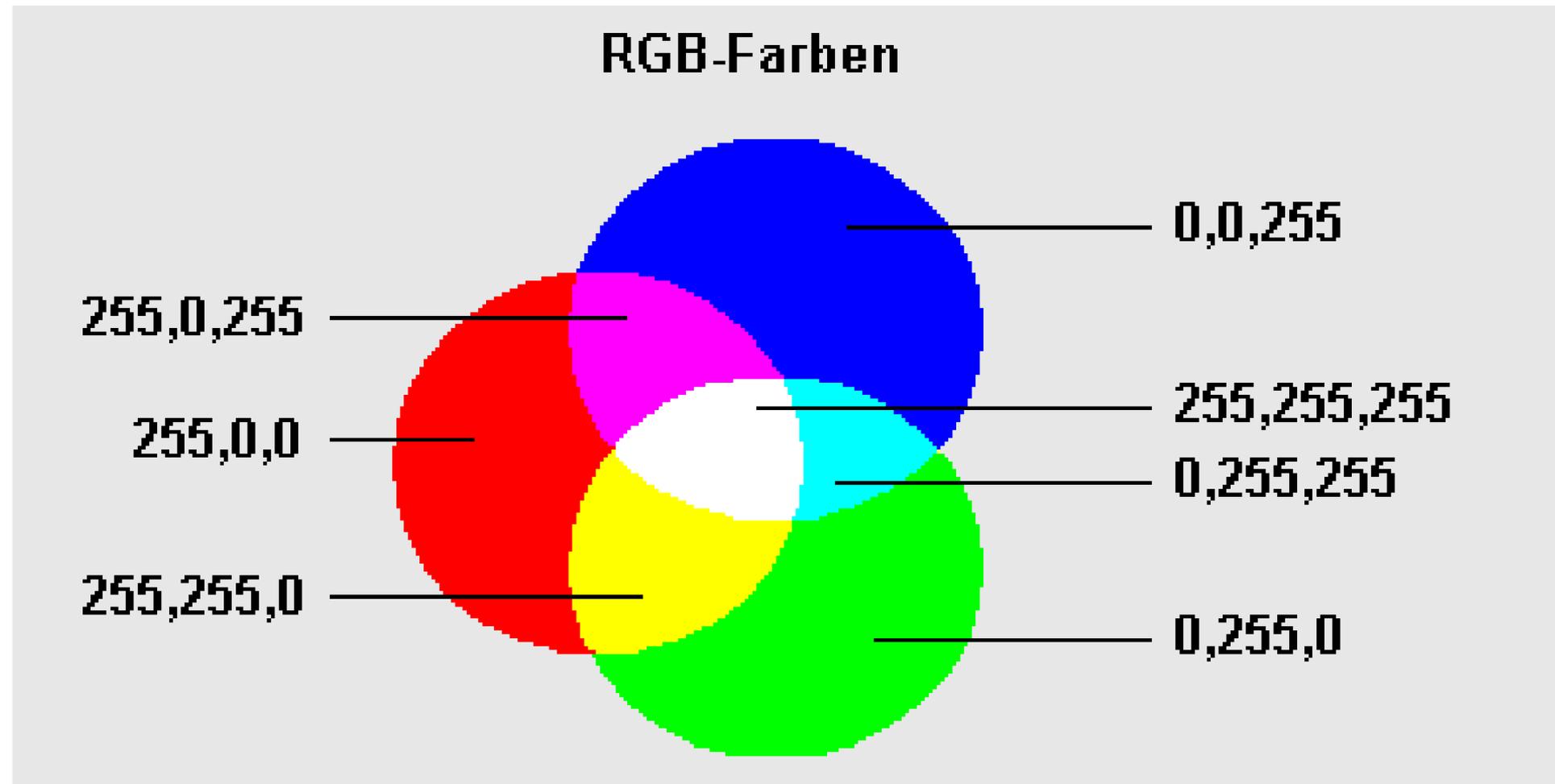
- Basis ist eine HTML-Seite als logische Sequenz von Blöcken.
- jeder Block wird mit einem passenden Tag gekennzeichnet
  - wie `<h1>...</h1>` oder
  - mit `<div>...</div>`
- Elemente, die speziell formatiert werden sollen, aber keinen Block erzeugen sollen, werden durch `<span>...</span>` gekennzeichnet.
- Die Reihenfolge entspricht der logischen Reihenfolge.
- Die Blöcke werden nun mit CSS formatiert, positioniert und ausgerichtet.

# Farben

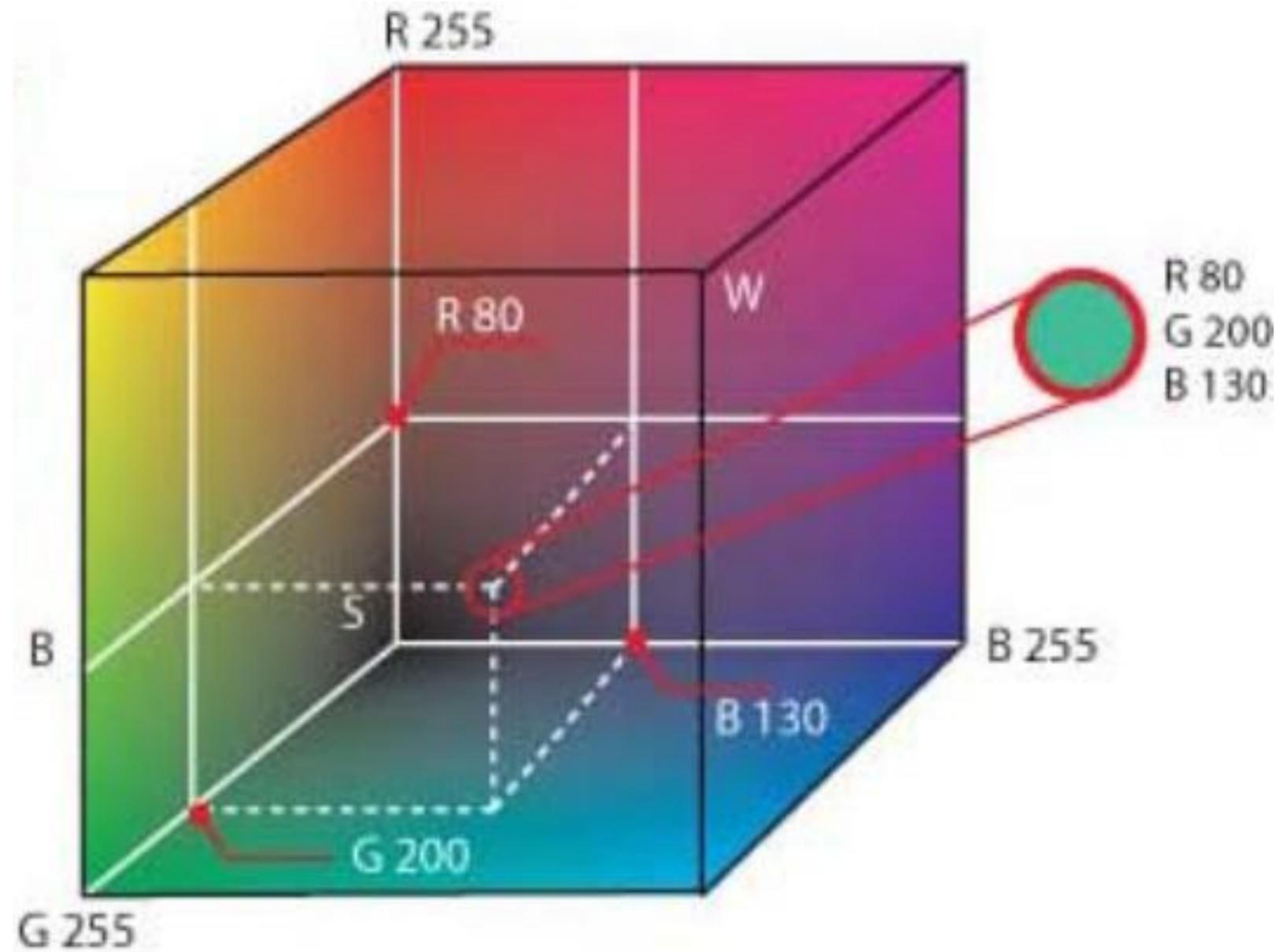
# Der RGB-Farbraum

- Beim RGB-Modell wird eine Farbe durch ihre Anteile an den drei Grundfarben Rot, Grün und Blau definiert.
- Jede Farbe hat also einen Rotwert, einen Grünwert und einen Blauwert.
- Jeder der drei Werte wird durch Zahlen zwischen 0 und 255 definiert.
- Der Wert 0 bedeutet:
  - Kein Anteil an der betreffenden Grundfarbe.
- Der Wert 255 bedeutet:
  - Maximaler Anteil an der betreffenden Grundfarbe.
- Die folgende Abbildung zeigt, wie Farbtöne beim RGB-Modell durch addierendes Mischen zustande kommen...

# Der RGB-Farbraum



# Der RGB-Farbraum



# Der RGB-Farbraum: HEX-Schreibweise

- Die hexadezimale Schreibweise ist die am weitesten verbreitete Methode, einen Farbwert anzugeben.
- Nach einem Gatterzeichen (hash) werden die Farbanteile der Farben rot, grün und blau in einem Bereich von 00 (0) bis FF (255) notiert.
  - Allgemein: #RRGGBB.
- CSS erlaubt eine Kurzschreibweise.
  - Allgemein: #RGB.
  - Die Kurzschreibweise ist nur dann möglich, wenn die erste und die zweite Ziffer aller Farbwerte identisch sind.

Farbe	Hexadezimaler Wert	Hinweis
	#ff7f50	Entspricht der benannten Farbe „coral“
	#abc	Entspricht #aabbcc
	#71f39b	Beliebiger Farbwert
	#000	Entspricht #000000

# Der RGB-Farbraum: Alternative Schreibweisen

- #FF8C00 - Farbanteile hexadezimal (00...FF)
- rgb(255,140,0) - Farbanteile rot, grün, blau im Bereich 0...255
- rgb(100%,55%,0%) - Farbanteile im Bereich 0%...100%
- hsl(0, 100%, 50%) - HSL (hue saturation lightness) Farbraum
- Angabe eines Alpha-Kanals als vierten Wert und somit eine Definition für eine Transparenz
- benannte Farben: black, white, olive, ...

# Einbinden in HTML

# Formate direkt in einem style-Attribut festlegen

- Mit dem Universalattribut „style“ können Sie Eigenschaften einem HTML-Element direkt zuweisen.
- Dabei sind nur Deklarationen, also Eigenschaft-Wert-Zuweisungen möglich.
- Durch das direkte Festlegen von Formaten, umgangssprachlich auch „Inline-Style“ genannt, gehen viele Vorteile verloren.
  - Der Wartungsaufwand steigt während sich die Flexibilität verringert.
  - Inline-Styles sind an ein Element gebunden und können nicht an zentraler Stelle bearbeitet werden.
  - Die Inline-Styles widersprechen auch der strikten Trennung von
    - Content in HTML5 und
    - Darstellung in CSS3.

# Formate direkt in einem style-Attribut festlegen

```
<body>  
  <h1 style="color: red;">Rote Überschrift</h1>  
</body>
```

- In diesem Beispiel wird die Überschrift „Rote Überschrift“ in der Schriftfarbe rot dargestellt.
- Prinzipiell ist es bei jedem HTML-Element möglich, direkt mit dem Attribut „style“ CSS-Formatierungen vorzunehmen.
- Dafür wird in der Attributliste des Elements das Attribut „style“ hinzugefügt
- In Anführungszeichen erfolgen dann die CSS-Angaben.
  - Diese müssen jeweils mit einem Semikolon voneinander getrennt werden
  - Leerzeichen sind erlaubt und dienen der Lesbarkeit

# Formate zentral in einem style-Element festlegen

- Das HTML-Element `style` erlaubt es, Formate zentral als Kindelement im `head` des HTML-Dokuments festzulegen.
- Im folgenden Beispiel wird das HTML-Element `<h1>` mit CSS adressiert und die Schriftfarbe in rot geändert.

```
<!DOCTYPE html>
<html lang="de">
  <head>
    . . .
    <style>
      h1 {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>Rote Überschrift</h1>
  </body>
</html>
```

# Einbinden einer externen Datei

- Im Normalfall besteht eine Webpräsenz aus oft hunderten von Seiten, die alle gleich formatiert werden sollen.
- Hier hat es sich durchgesetzt, das CSS in einem eigenen externen Stylesheet mit der Endung `.css` abzuspeichern und mit dem HTML-Element **link** direkt im head einzubinden.
- Folgende Attribute sind erforderlich und ausreichend, um ein Stylesheet einzubinden:
  - **rel** definiert den logischen Beziehungstyp des Elements.
    - „stylesheet“ bedeutet, dass ein Stylesheet eingebunden werden soll.
  - **href** referenziert die einzubindende Stylesheet-Datei.

# Einbinden einer externen Datei

- Das link-Element kann aber noch weitere Attribute besitzen, die auf Stylesheets eine Wirkung haben.
  - type enthält den Medientyp der einzubindenden Datei, für CSS also „text/css“. Da dies in HTML der Standard Typ ist, kann es weggelassen werden.
  - media kann festlegen, dass ein Stylesheet nur dann verarbeitet werden soll, wenn das Dokument von einem bestimmten Ausgabemedium ausgegeben wird.
    - Als Attributwert ist eine Medienabfrage erlaubt, beispielsweise screen für Bildschirme oder print für Drucker.
    - Ist das Attribut nicht definiert, gilt der Standardwert all.
    - CSS erlaubt es, mit Media Queries für Ausgabemedien mit bestimmten Eigenschaften, etwa Tablets, Smartphones aber auch Drucker abweichende Darstellungen festzulegen.

# Einbinden einer externen Datei

```
<!DOCTYPE html>
<html lang="de">
  <head>
    . . .
    <link rel="stylesheet" type="text/css" href="design.css">
  </head>
  <body>
    <h1>Rote Überschrift</h1>
  </body>
</html>
```

- In diesem Beispiel wird eine Datei mit dem Namen „design.css“ in das HTML-Dokument inkludiert.
- Dabei werden alle in dieser Datei enthaltenen Styles auf das HTML-Dokument angewendet.