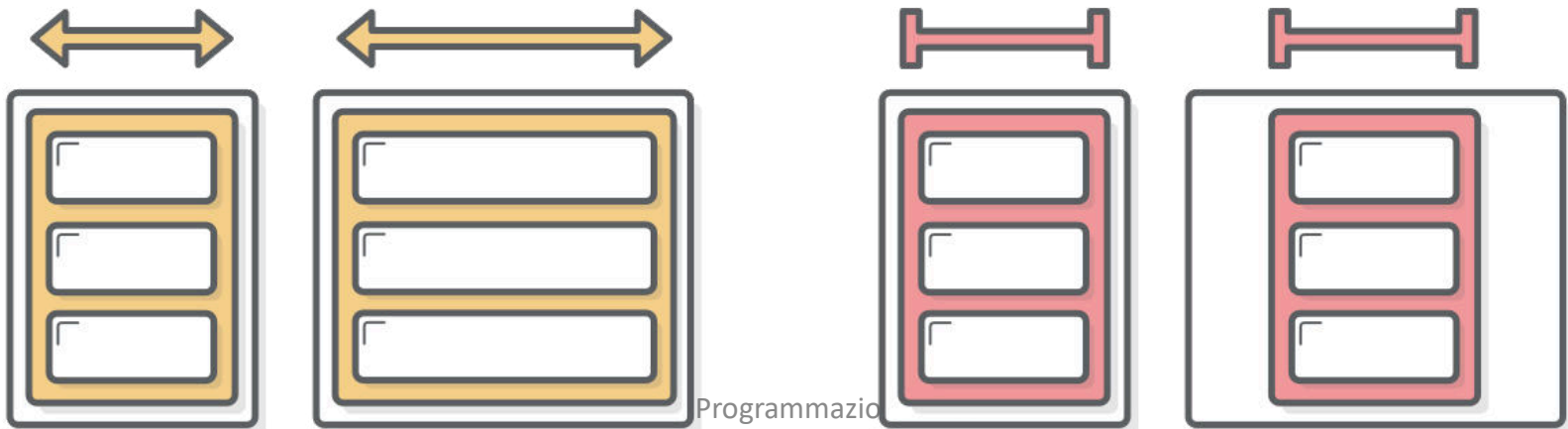


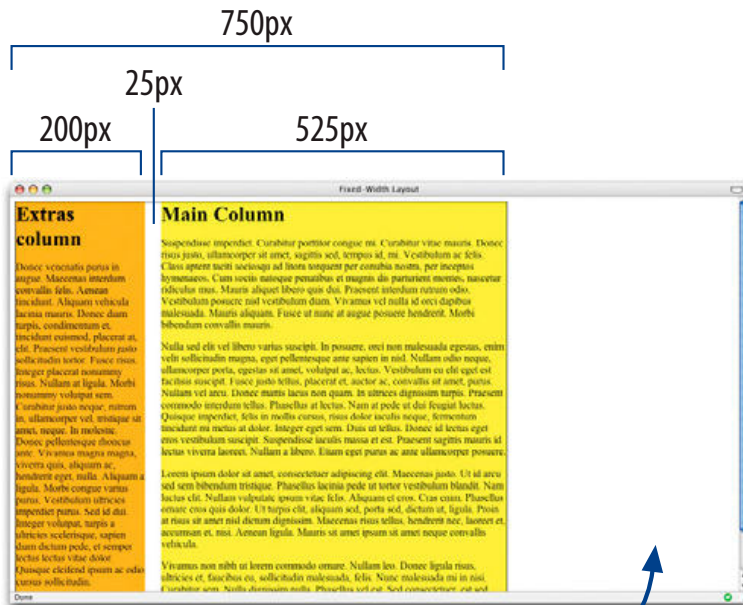
# STRUTTURA DELLA PAGINA

# Layout delle pagine

- **Fixed**
  - dimensione in pixel fissa
- **Fluid**
  - proporzionale alla larghezza del browser
- **Elastic**
  - Proporzionale al testo
- **Hybrid**
  - Misto fra fixed e fluid

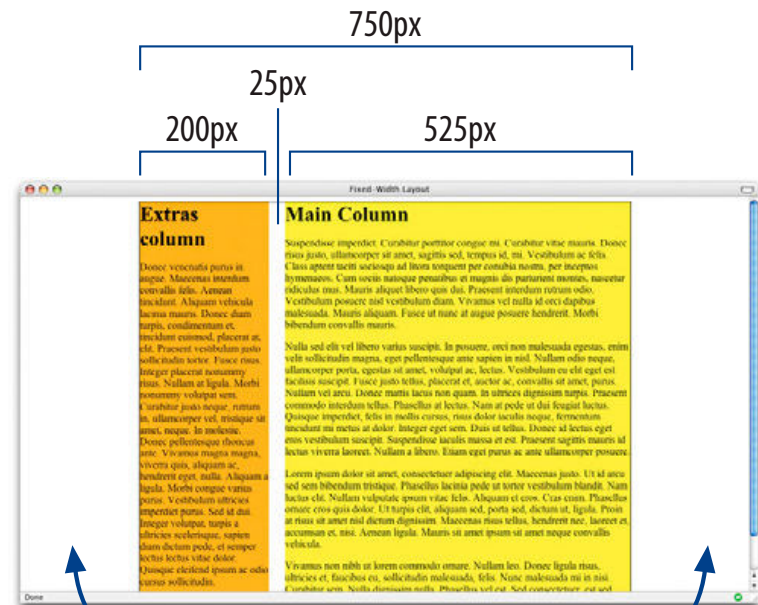


# Fixed layouts



Extra space on right

```
#wrapper {width: 750px;
position: absolute;
margin-left: auto;
margin-right: auto;
border: 1px solid black;
padding: 0px;}
```



Extra space split on left and right sides

```
#extras {position: absolute;
top: 0px;
left: 0px;
width: 200px;
background: orange; }

#main {margin-left: 225px;
background-color: yellow;}
```

# Discussione Layout Fisso

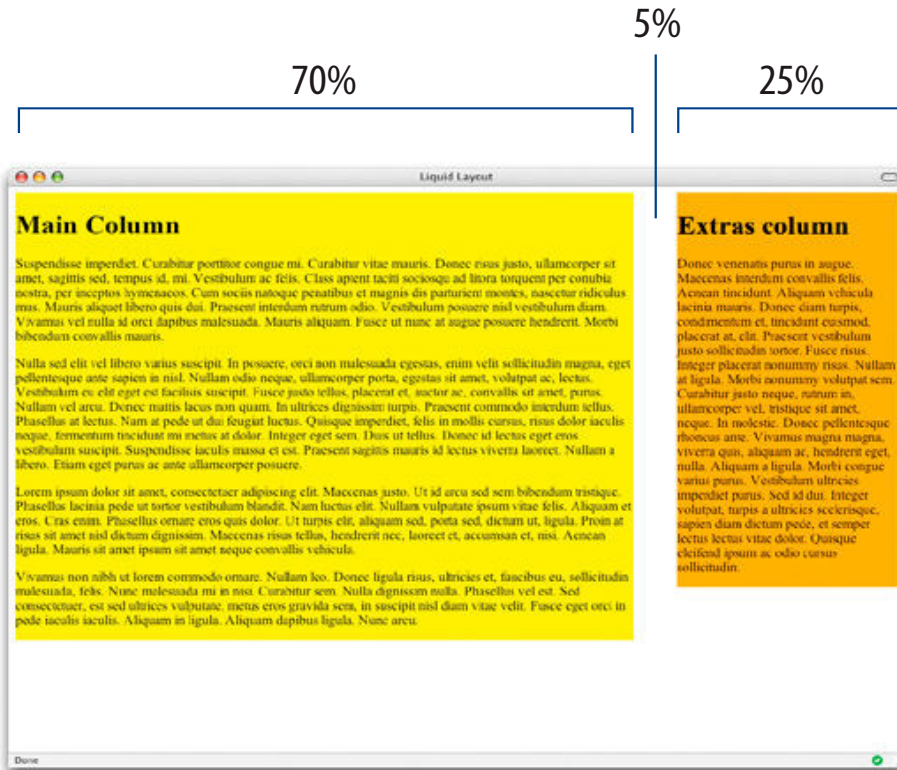
- **Vantaggi**

- Controllo del numero delle righe
- è semplice da realizzare
- è una visualizzazione comune per i desktop

- **Svantaggi**

- se la finestra è piccola la parte destra viene coperta
- se lo schermo è grande ho spazi a dx ed sx da gestire
- se ho carattere grandi ho problemi di righe corte
- la pagina non è controllabile dall'utente

# Fluid layout



```
div#main {
width: 70%;
margin-right: 5%;
float: left;
background: yellow;
}
```

```
div#extras {
width: 25%;
float: left;
background: orange;
}
```

# Discussione Layout Fluido

- **Vantaggi**

- Sembra più vicino alla natura dei browser
- Elimina spazi potenzialmente vuoti
- L'utente può controllare la pagina
- Non ho scrollbar orizzontali

- **Svantaggi**

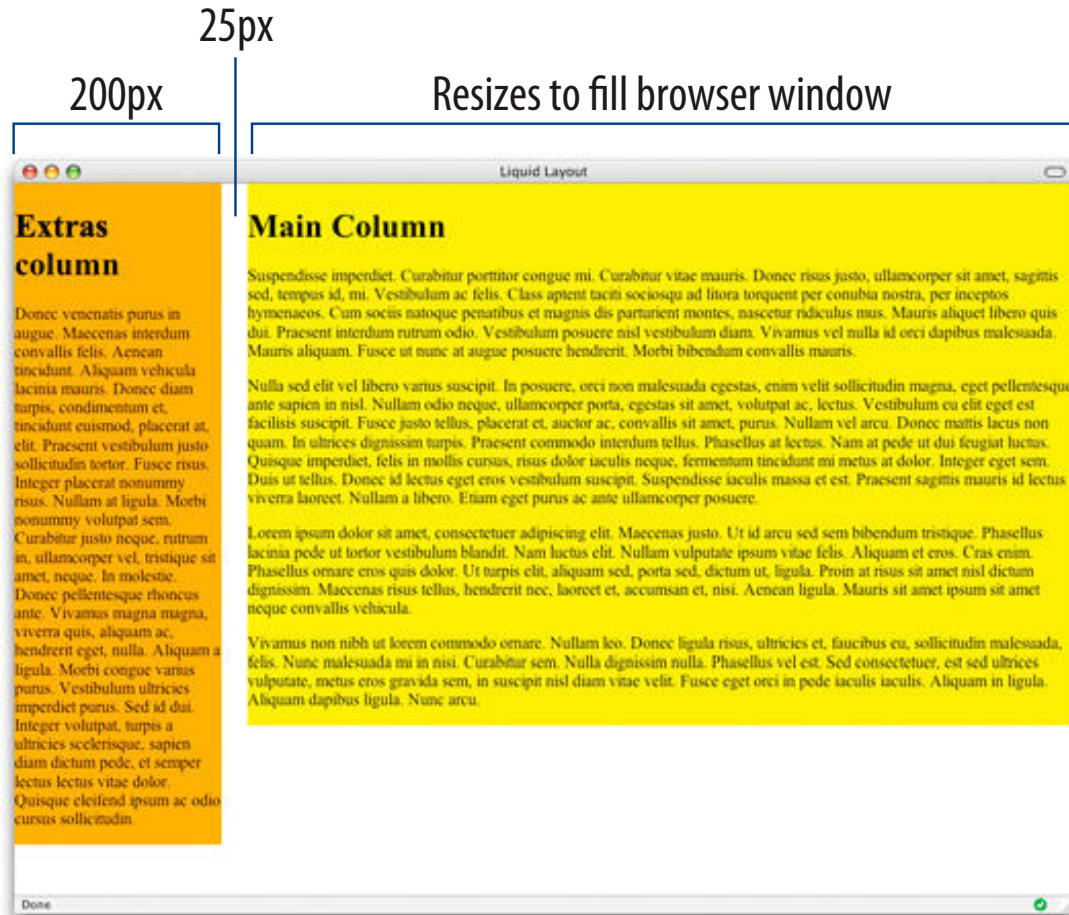
- Se il monitor è grande ho righe troppo lunghe
  - limitare la dimensione
- Non è facile predire la posizione degli elementi e l'effetto grafico finale
- Problemi nei browser piccoli
- I conti sono un poco più complicati



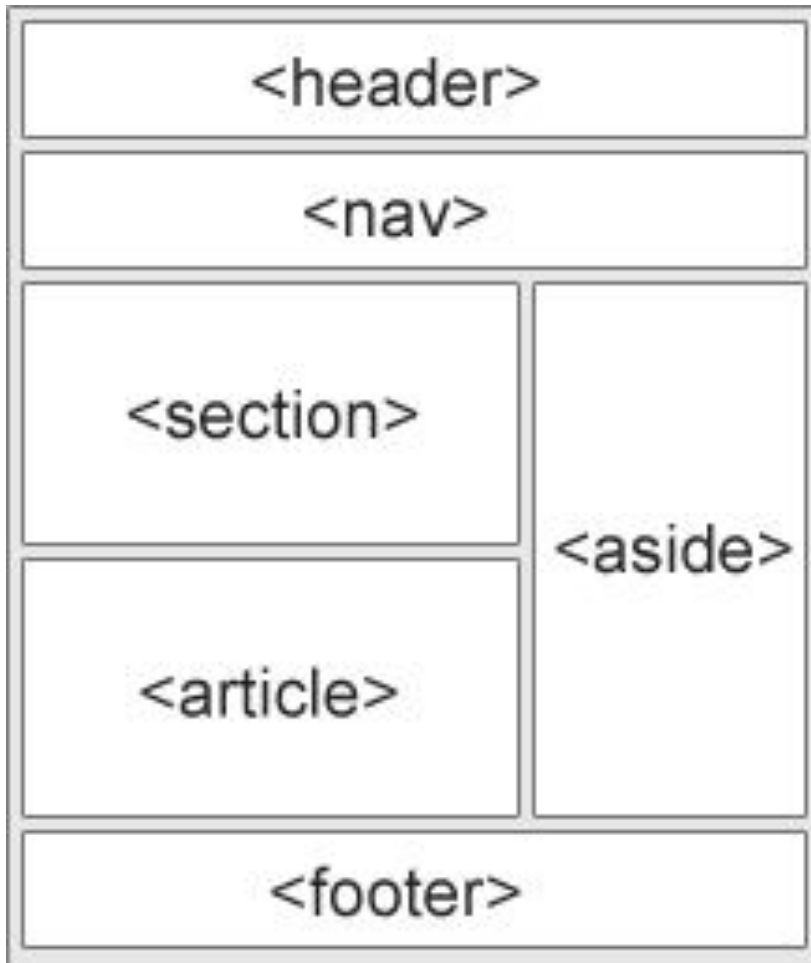
# Layout ibrido

```
div#main {  
  width: auto;  
  position: absolute;  
  top: 0;  
  left: 225px;  
  background: yellow; }
```

```
div#extras {  
  width: 200px;  
  position: absolute;  
  top: 0;  
  left: 0;  
  background: orange; }
```



# Struttura pagina Desktop

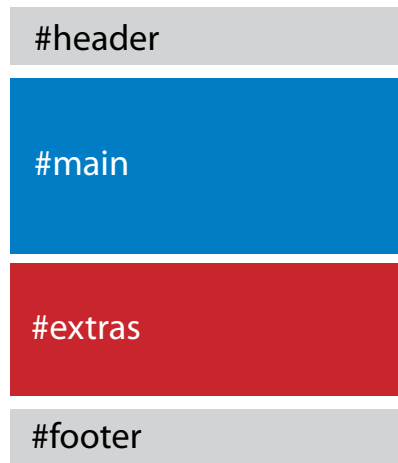




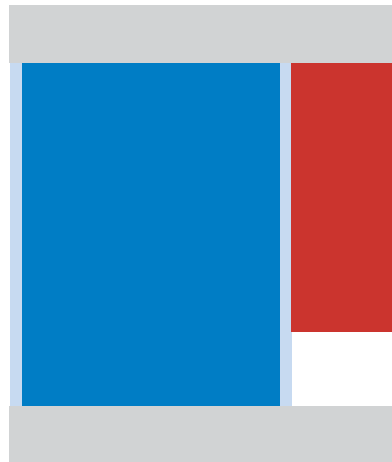
# Two columns, fluid layout

```
<div id="header">Masthead and headline</div>
<div id="main">Main article</div>
<div id="extras">List of links and news</div>
<div id="footer">Copyright information</div>
```

```
#main {
  float: left;
  width: 60%;
  margin: 0 5%;
}
#extras {
  float: left;
  width: 25%;
  margin: 0 5% 0 0;
}
#footer {
  clear: left;
}
```



Source order



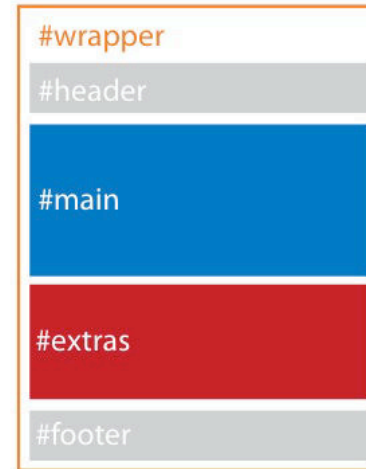
Layout



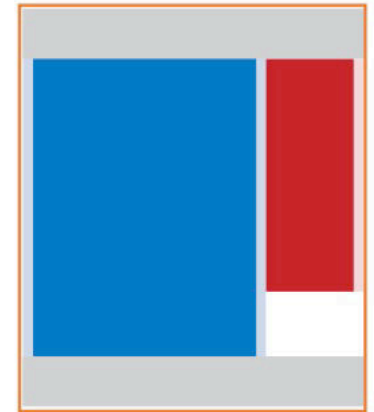
# Two columns, fixed-width layout

```
<div id="wrapper">
  <div id="header">Masthead and headline</div>
  <div id="main">Main article</div>
  <div id="extras">List of links and news</div>
  <div id="footer">Copyright information</div>
</div>
```

```
#wrapper {
  width: 960px;
}
#main {
  float: left;
  width: 650px;
  margin: 0 20px;
}
#extras {
  float: left;
  width: 250px;
  margin: 0 20px 0 0;
}
#footer {
  clear: left;
}
```



Source order

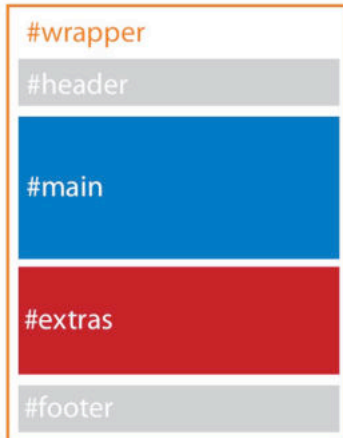


Layout

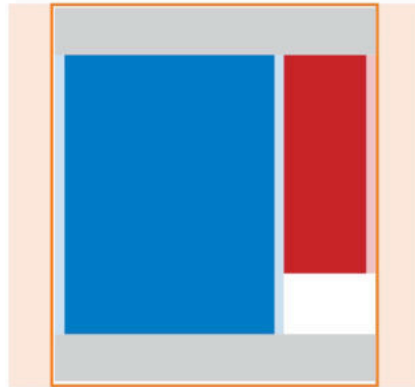


# Two columns, fixed-width, centered

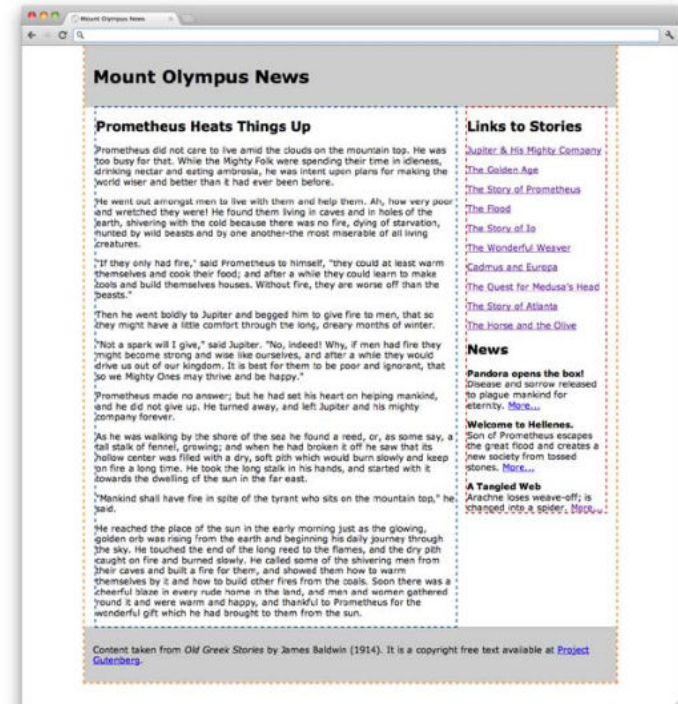
```
#wrapper {
  width: 960px;
  margin: 0 auto;
}
```



Source order

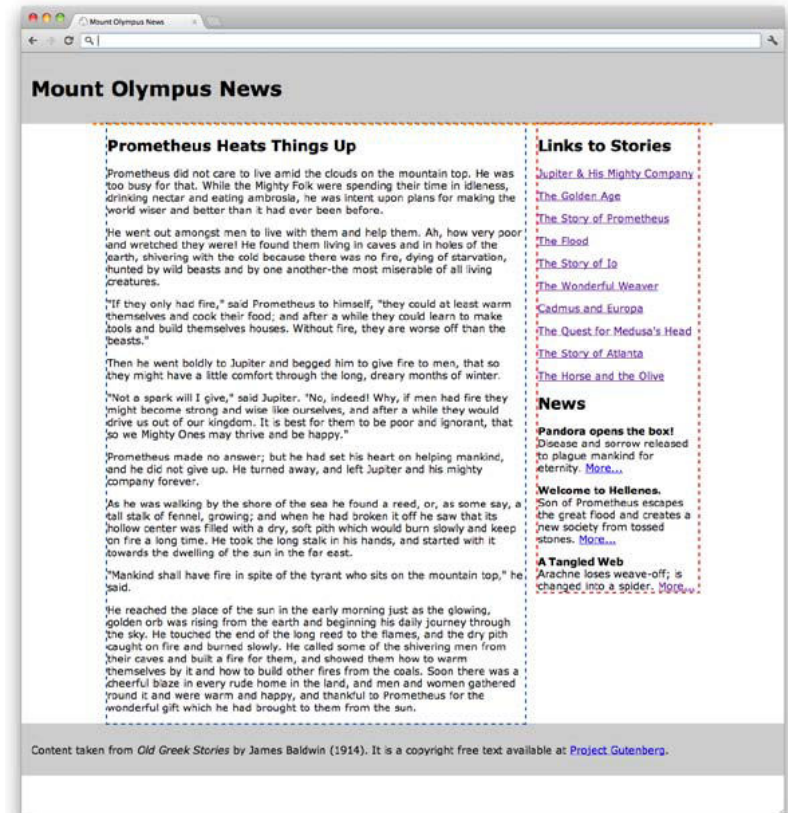


Layout



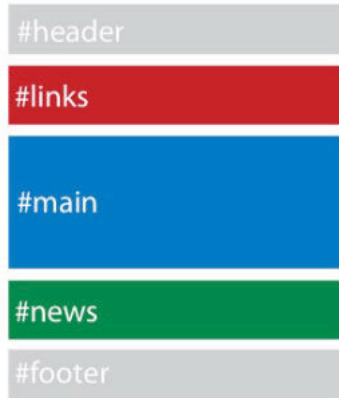
# header e footer full width

```
<div id="header">Masthead and headline</div>
<div id="wrapper">
  <div id="main">Main article</div>
  <div id="extras">List of links and news</div>
</div>
<div id="footer">Copyright information</div>
```



# Three columns, fluid layout

```
#links {
  float: left;
  width: 22.5%;
  margin: 0 0 0 2.5%;
}
#main {
  float: left;
  width: 45%;
  margin: 0 2.5%;
}
#news {
  float: left;
  width: 22.5%;
  margin: 0 2.5% 0 0;
}
#footer {
  clear: left;
}
```



Source order



Layout

```
<div id="header">Masthead and headline</div>
<div id="links">List of links</div>
<div id="main">Main article</div>
<div id="news">News items</div>
<div id="footer">Copyright information</div>
```



```
#wrapper {
    width: 960px;
    margin: 0 auto;
}

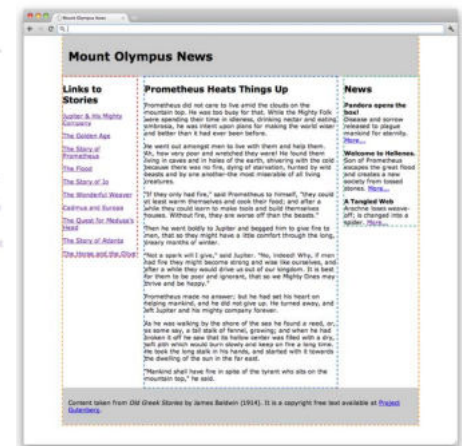
#main {
    float: left;
    width: 520px;
    margin-top: 0;
    margin-left: 220px;
    margin-right: 20px;
}

#news {
    float: left;
    width: 200px;
    margin: 0;
}
```

A vertical stack of six colored rectangles representing web page sections. From top to bottom: a light gray rectangle labeled #wrapper, a light gray rectangle labeled #header, a large blue rectangle labeled #main, a green rectangle labeled #news, a red rectangle labeled #links, and a light gray rectangle labeled #footer.

A diagram of a 2D array represented as a grid of colored squares. The grid is 3 rows high and 3 columns wide. The top row contains a red square, a blue square, and a green square. The middle row contains a light blue square, a blue square, and a light green square. The bottom row contains a light blue square, a blue square, and a light green square. A white arrow points from the right side of the red square to the left side of the blue square in the middle row. Another white arrow points from the right side of the light blue square in the bottom row to the left side of the blue square in the bottom row. This illustrates row-major traversal, where elements are visited row by row.

Progr



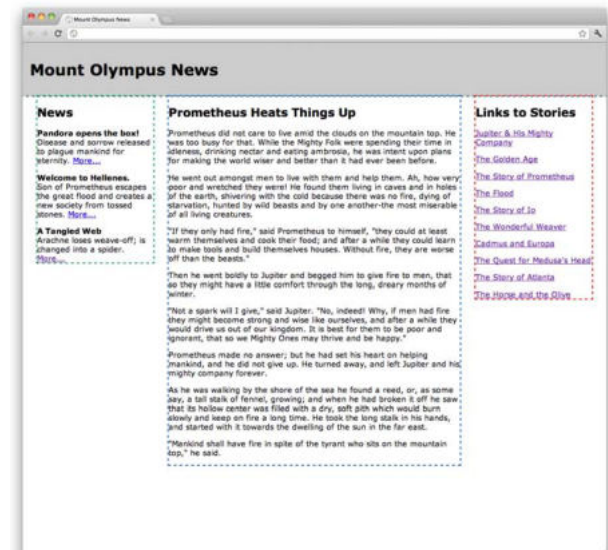
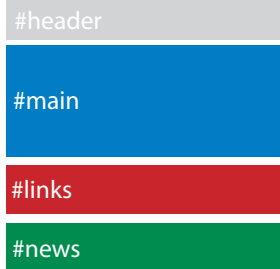


# Posizionamento absolute fluid

```
<div id="header">Masthead and headline</div>
<div id="content">
  <div id="main">Main article</div>
  <div id="news">News items</div>
  <div id="links">List of links</div>
</div>
```

```
#content {
  position: relative;
  margin: 0;
}
#main {
  width: 50%;
  position: absolute;
  top: 0;
  left: 25%;
  margin: 0;
}
```

```
#news {
  width: 20%;
  position: absolute;
  top: 0;
  left: 2.5%;
  margin: 0;
}
#links {
  width: 20%;
  position: absolute;
  top: 0;
  right: 2.5%;
  margin: 0;
}
```





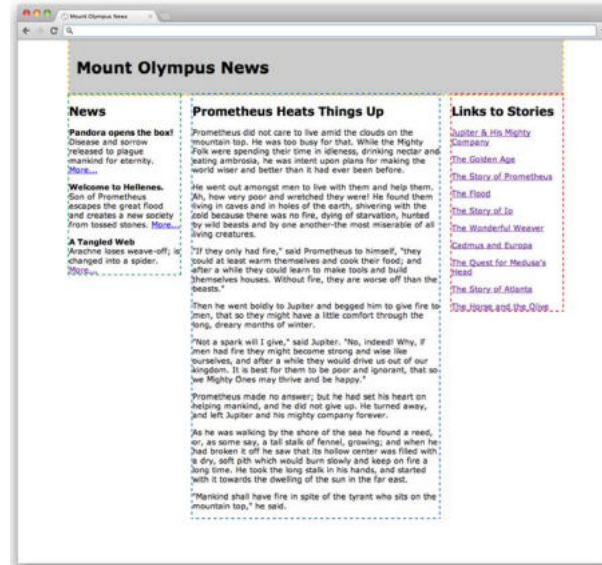
# Posizionamento absolute fixed



Source order



Layout



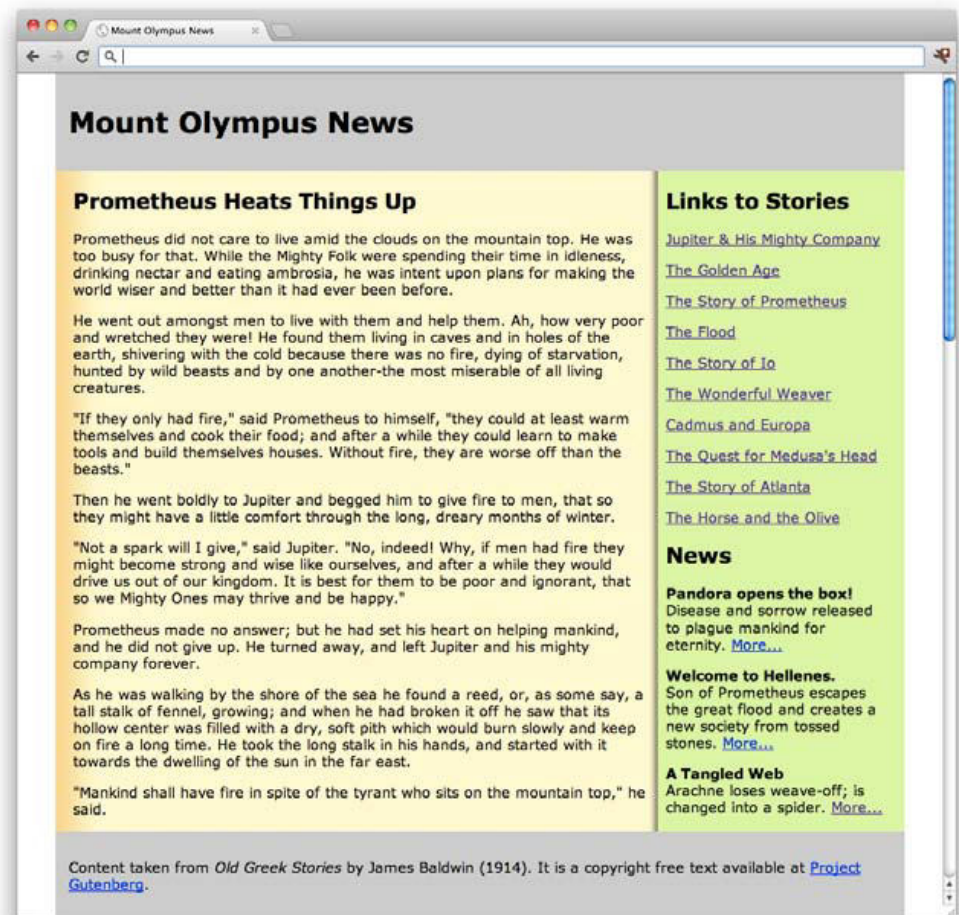
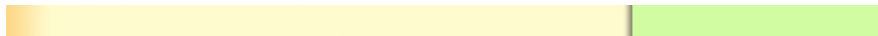
```
#wrapper {
  width: 960px;
  margin: 0 auto;
}
#content {
  margin: 0;
  position: relative;
}
#main {
  width: 520px;
  position: absolute;
  top: 0;
  left: 220px;
  margin: 0;
}
```

```
#news {
  width: 200px;
  position: absolute;
  top: 0;
  left: 0;
  margin: 0;
}
#links {
  width: 200px;
  position: absolute;
  top: 0;
  right: 0;
  margin: 0;
}
```

# Tecniche per Background fixed layout

```
#wrapper {  
  width: 960px;  
  margin: 0 auto;  
  background-image: url(two_column.gif);  
  background-repeat: repeat-y;  
}
```

two\_columns.png



# Tecniche per Background fluid layout

```
body {
  background-image: url(two_cols_3000px);
  background-repeat: repeat-y;
  background-position: 67.5%;
}
```

*two\_cols\_3000px.png*

background-position: 67.5%



# MEDIA QUERY

# Media query

- Definisco degli stili per determinati media e device

```
@media not|only mediatype and (media feature) {  
    CSS-Code;  
}
```

```
<link rel="stylesheet" media="mediatype and|not|only (media feature)" href="mystylesheet.css">
```

# Media Type

- all, print, screen, speech

```
@media screen {  
  p {  
    font-family: verdana, sans-serif;  
    font-size: 17px;  
  }  
}
```

```
@media print {  
  p {  
    font-family: georgia, serif;  
    font-size: 14px;  
    color: blue;  
  }  
}
```

```
@media screen, print { ... }
```

# Media Features

@media (orientation: landscape) { ... }

@media (max-width: 12450px) { ... }

@media (color) { ... }



# Media features

Feature	Description
width	The width of the display area (viewport).
height	The height of the display area (viewport).
device-width	The width of the devices rendering surface (the whole screen).
device-height	The height of the devices rendering surface (the whole screen).
orientation	Whether the device is in portrait or landscape orientation. (Does not accept min-/max- prefixes.)
aspect-ratio	Ratio of the viewport's width divided by height (width/height).
device-aspect-ratio	Ratio of the whole screen's (rendering surface) width to height.
color	The bit depth of the display; for example, color: 8 tests for whether the device has at least 8-bit color.
color-index	The number of colors in the color lookup table.
monochrome	The number of bits per pixel in a monochrome device.
resolution	The density of pixels in the device. This is increasingly relevant for detecting high-resolution displays.
scan	Whether a tv media type uses progressive or interlace scanning. (Does not accept min-/max- prefixes.)
grid	Whether the device uses a grid-based display, such as a fixed-width font. (Does not accept min-/max- prefixes.)

[https://www.w3schools.com/cssref/css3\\_pr\\_mediaquery.asp](https://www.w3schools.com/cssref/css3_pr_mediaquery.asp)

# Media Query complesse

- Combinare le features

- @media (min-width: 30em) **and** (orientation: landscape) { ... }
- @media screen **and** (min-width: 30em) **and** (orientation: landscape) { ... }
- @media print **and** (min-resolution: 300dpi) { ...styles here... }

- Combinare le query

- @media (min-height: 680px), screen and (orientation: portrait) { ... }
-

# Esempi MQ

- [https://www.w3schools.com/csS/css3\\_mediaqueries.asp](https://www.w3schools.com/csS/css3_mediaqueries.asp)
- [https://www.w3schools.com/csS/css3\\_mediaqueries\\_ex.asp](https://www.w3schools.com/csS/css3_mediaqueries_ex.asp)
- [https://www.w3schools.com/cssref/tryit.asp?filename=trycss3\\_media2](https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_media2)