

BESSERE FORMULARE

BEKANNTE FORMULARELEMENTE

HTML-Code

`<form action=“„ method=“„>`

Beschreibt ein Formular.

`<fieldset>`

Beschreibt eine Sammlung von Eingabefeldern.

`<input type=“text“>`

Einfaches Eingabefeld

`<input type=“password“>`

Eingabefeld für für Passworteingaben

`<input type=“submit“>`

Eingabefeld zum Abschicken des Formulars

`<label>`

Beschreibung für ein Eingabefeld

AUFBAU EINES FORMULARS

HTML

```
<h1>Kontaktformular</h1>

<form action="register.php" method="post">

  <fieldset>

    <label for="vorname">Vorname:</label>
    <input type="text" name="vorname" id="vorname">

    <label for="nachname">Nachname:</label>
    <input type="text" name="nachname" id="nachname">

    <label for="email">Email-Adresse:</label>
    <input type="text" name="email" id="email">

    <textarea name="nachricht" id="nachricht">
    </textarea>

    <input type="submit">

  </fieldset>

</form>
```

KONTAKTFORMULAR

VORNAME:	<input type="text"/>
NACHNAME:	<input type="text"/>
EMAIL-ADRESSE:	<input type="text"/>
	<input type="text"/>
<input type="submit" value="Submit"/>	

NEUE TYPEN FÜR EINGABEN

Zur Erfassung von Benutzereingaben stehen bisher die Elemente `<input>` und `<textarea>` zur Verfügung. Durch Definition des Attributs „**type**“ wird die Art der Eingabe spezifiziert:

HTML-Code

```
<input type="text" id="vorname" name="vorname">
```

Bisher waren nur wenige bzw. sehr allgemein formulierte Werte für „**type**“ möglich. So beispielsweise „**text**“ oder „**password**“.

Durch in HTML5 neu eingeführte zusätzliche Typen wird eine genauere Beschreibung der Eingabefelder möglich. Dies ermöglicht dem Browser eine bessere Handhabung der Eingaben:

- Anzeige passender virtueller Tastaturen auf mobilen Endgeräten
- Einblendung von Eingabehilfen wie z. B. Spinnboxen oder Datumswählern

EINSATZ DER NEUEN TYPEN

Probleme

- aktuell schlechte Browserunterstützung
- unterschiedliche Darstellung in verschiedenen Browsern

Lösungsansätze

- trotzdem verwenden weil zukunftssicher
- Ausweichlösungen mit JavaScript selbst entwickeln
- Ausweichlösung mit Modernizr

Für viele der neuen Formulartypen sind Fallbacks nicht unbedingt notwendig. Kennt der Browser einen Typen nicht, ersetzt er denn Typen meistens durch den Wert „text“. Dadurch wird eine einfaches Eingabefeld dargestellt.

Im Web existieren viele Quellen um die aktuelle Unterstützung neuer Elemente und Eigenschaften zu prüfen. Die Website <http://fmbip.com/litmus/> stellt eine solche Referenz zur Verfügung.

„EMAIL“, „URL“ UND „TEL“

In HTML5 halten Typen für Telefonnummern, Email- und Webadressen Einzug. Desktopbrowser stellen das gewohnte Eingabefeld dar. Hier ergibt sich weder in der Darstellung, noch in der Funktion eine Änderung.

Auf mobilen Endgeräten, die eine virtuelle Tastatur verwenden, kann die angezeigte Tastaturbelegung abhängig vom Typ verändert dargestellt werden.

HTML-Code

```
<input type="email">
```

```
<input type="url">
```

```
<input type="tel">
```



Veränderte Tastatur für den Typ „Email“ auf dem iPhone.

Quelle: <http://www.bennadel.com>

“SEARCH”

HTML-Code

```
<input type="search">
```

Eingabefeld für die Suche mit Schlüsselwörtern.
Dabei verhält sich das Eingabefeld wie ein reguläres Textfeld.

“RANGE“

Der Typ „range“ visualisiert einen Slider. Der User kann den Wert „via Drag & Drop“ verändern.

Minimum und Maximung des Feldes können über zwei zusätzliche Attribute definiert werden.

Der Startwert kann über das Attribut „value“ vorgegeben werden.

Zusätzlich kann mit dem Attribut „step“ ein Intervall für die Einzelschritte definiert werden.

HTML-Code

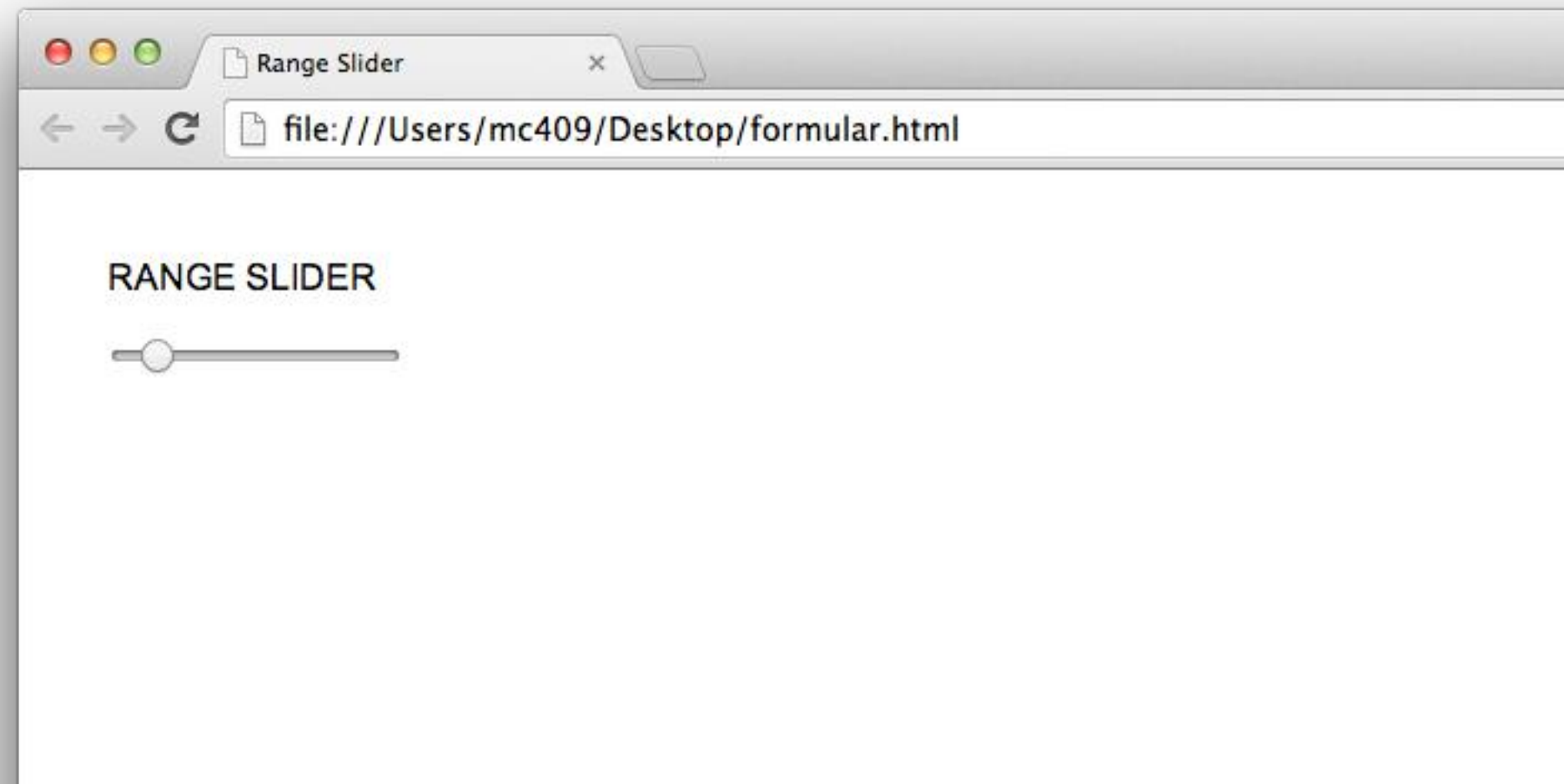
```
<input type="range" min="1" max="200" value="2">
```

Kleinst mögliche Zahl

Größt mögliche Zahl

Initialwert

DARSTELLUNG IN GOOGLE CHROME



“NUMBER“

Der Typ „number“ bietet einen Zahlenwähler an. Der User kann den Wert über die Schaltflächen nach oben und unten korrigieren.

Die Eingabe via Tastatur ist weiter möglich.

Wie beim Typ „range“ kann ein Minimal- und Maximalwert definiert werden. Auch die Angabe des Startwerts ist durch das Attribut „value“ möglich.

HTML-Code

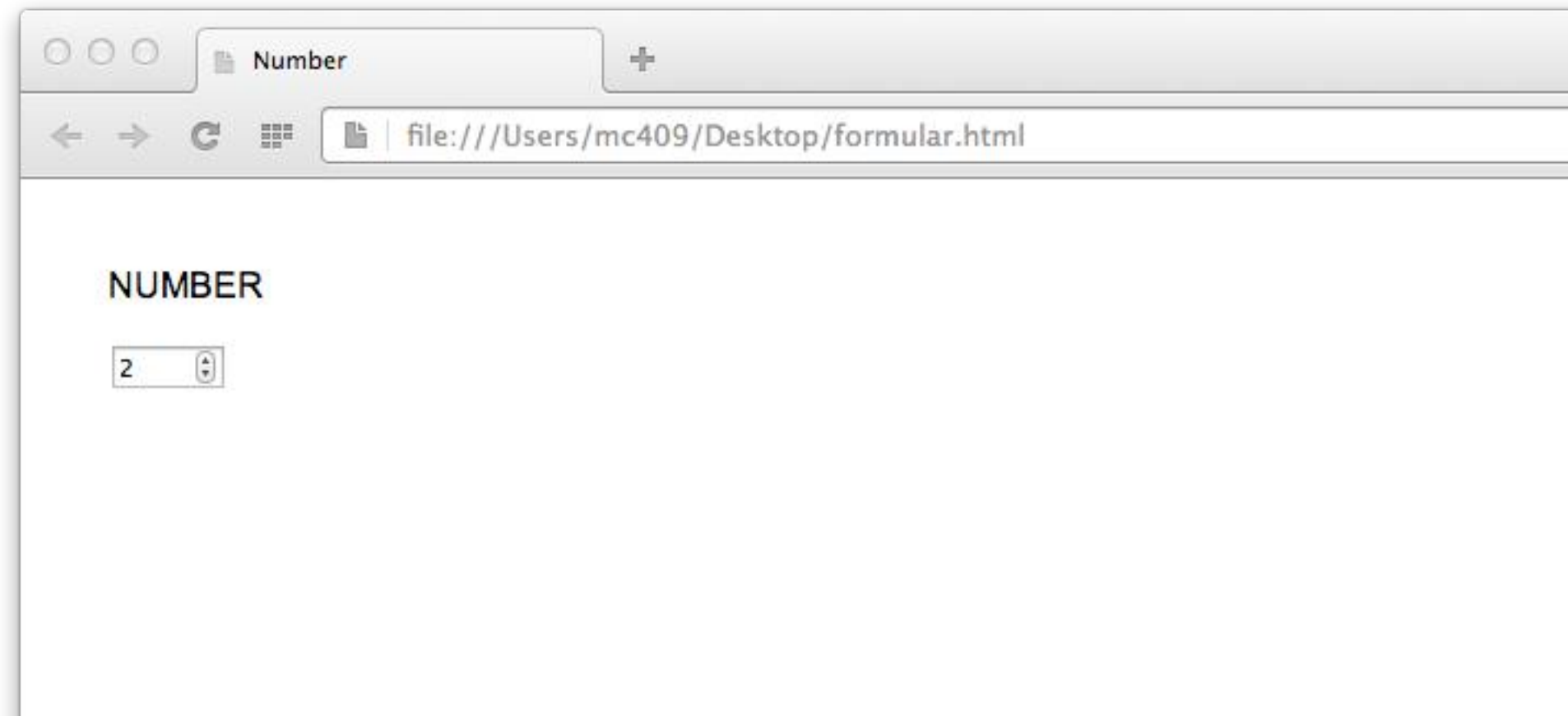
```
<input type="number" min="1" max="200" value="2">
```

Kleinst mögliche Zahl

Größt mögliche Zahl

Initialwert

DARSTELLUNG IN OPERA



“DATE“

Mit dem Typ „date“ kann ein Datumswähler realisiert werden. Der Benutzer kann den gewünschten Wert über einen visualisierten Kalender mit der Maus wählen.

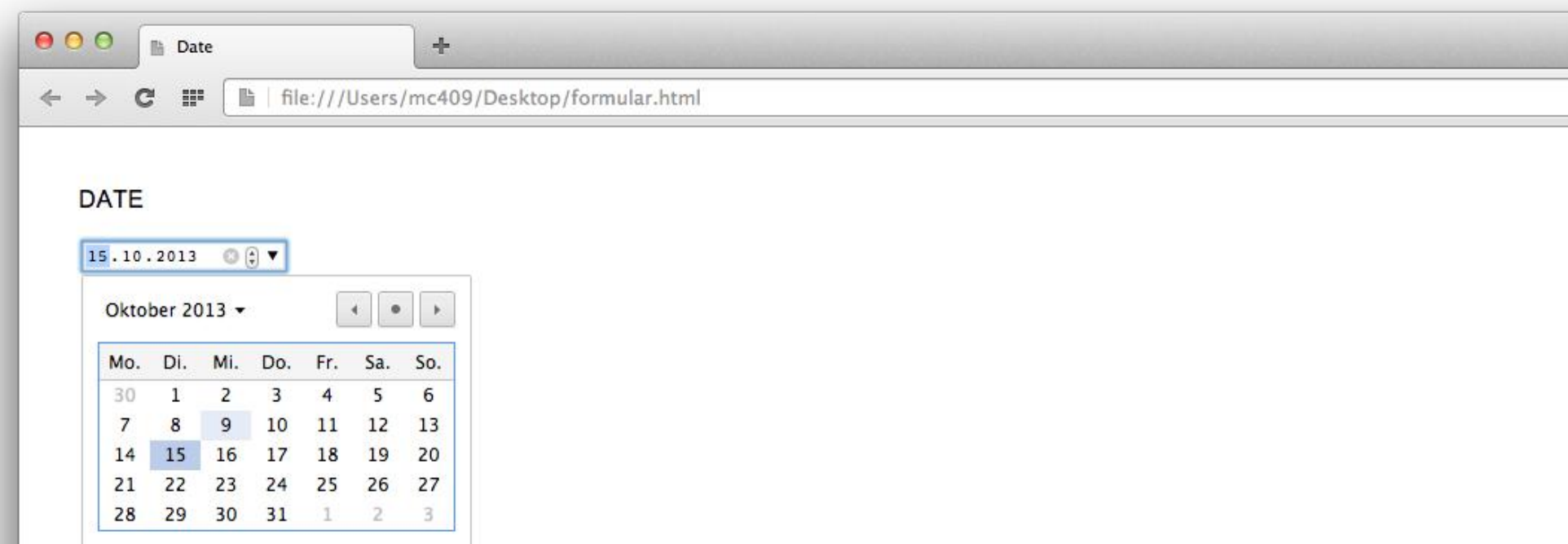
Bisher mussten Sie Datumswerte mit Hilfe regulärer Eingabefelder erfassen. Dabei besteht die Gefahr, dass Benutzer Ihre Eingaben in einem anderen Format als gewünscht erfassen. Mit einem sog. „Datepicker“ erhalten Sie jede Eingabe im gleichen Format.

Über das Attribut „value“ kann ein Startwert vorgegeben werden.

HTML-Code

```
<input type="date" value="2013-10-15">
```

DARSTELLUNG IN OPERA



„DATETIME“ UND „DATETIME-LOCAL“

Die beiden Attributwerte erweitern den von „date“ bekannten Datepicker um Möglichkeit eine Uhrzeit anzugeben.

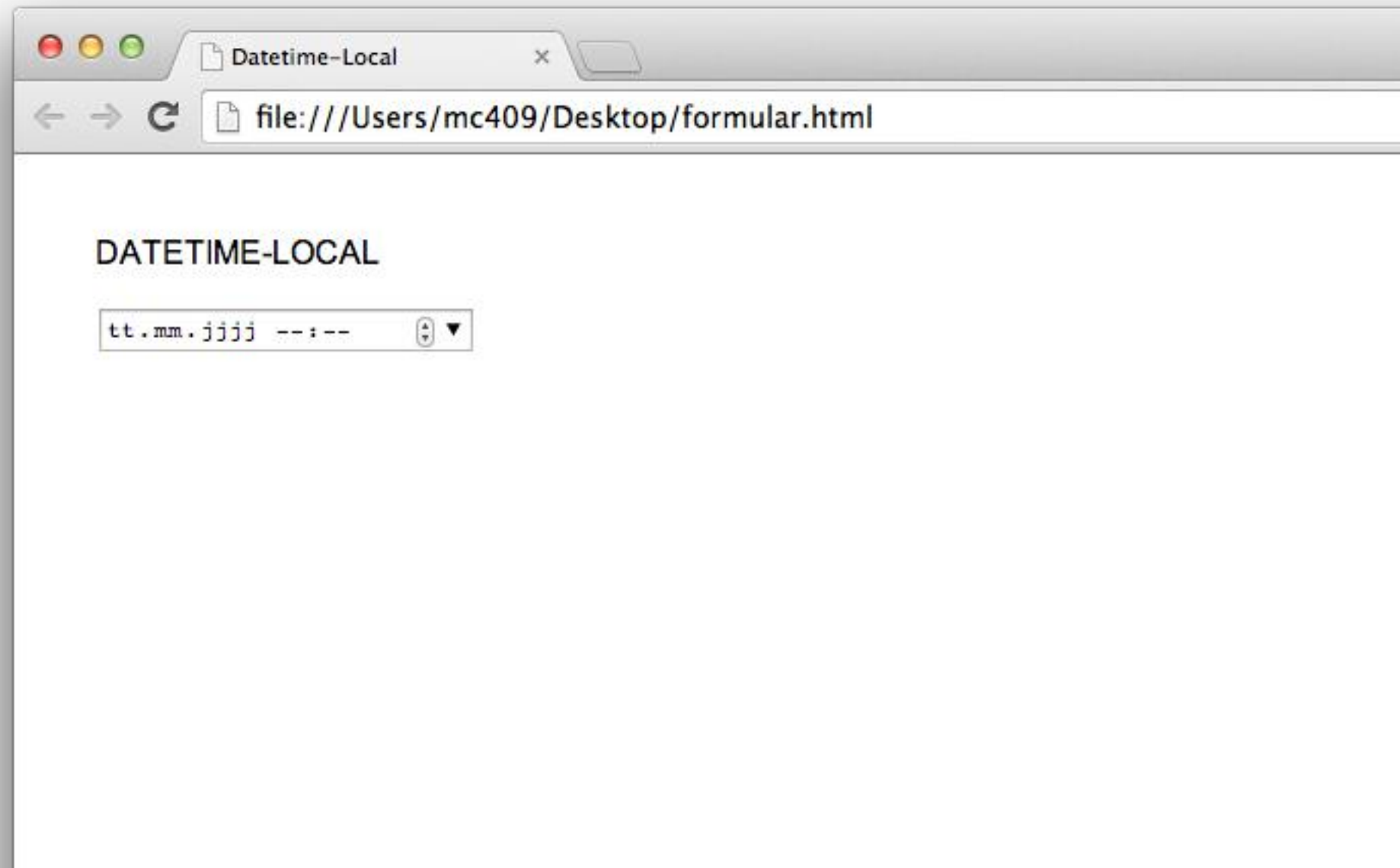
Dabei arbeitet „datetime“ mit einer Zeitzone und „datetime-local“ ohne.

Achtung! Offiziell bieten mehrere Browser Unterstützung für beide Attributwerte. Jedoch konnte nur „datetime-local“ zuverlässig getestet werden.

HTML-Code

```
<input type="datetime-local">
```

DARSTELLUNG IN GOOGLE CHROME



WEITERE DATUMSBEZOGENE ATTRIBUTWERTE

Neben den besprochenen Attributwerten existieren weitere datumbezogene Werte:

HTML-Code

```
<input type="month">
```

Liefert einen Datepicker, der nur Monat und Jahr verarbeitet

```
<input type="week">
```

Liefert einen Datepicker, der Woche und Jahr verarbeitet

```
<input type="time">
```

Zeigt ein Kontrollelement zur Eingabe der Uhrzeit (ohne Zeitzone)

„COLOR“

Der Attributwert ermöglicht einen Farbwähler. Aktuell wird dieser aber nur von einigen Browsern unterstützt. Diese stellen zunächst den gewählten Farbwert in einem kleinen Fenster dar.

Achtung!

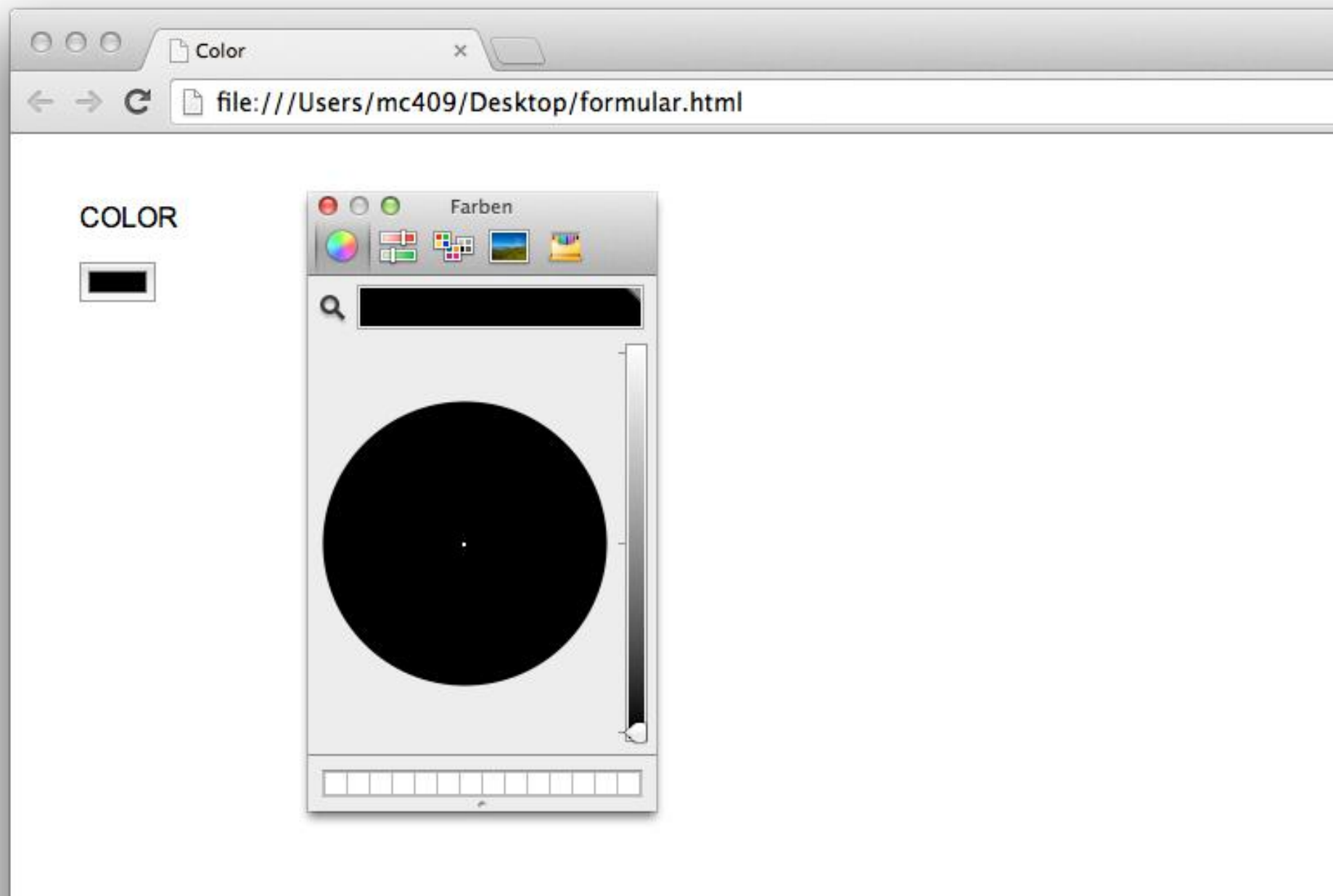
Auf Mac OS wird nach einem Klick auf die Schaltfläche ein Colorpicker angezeigt, der vom Betriebssystem zur Verfügung gestellt wird.

Um einen Colorpicker direkt im Browser zu erstellen müssten Sie hier den Weg via JavaScript bzw. jQuery wählen.

HTML-Code

```
<input type="color">
```

DARSTELLUNG IN GOOGLE CHROME



ÜBERBLICK DER NEUEN INPUT-TYPEN

„email“

„date“

„tel“

„datetime(-local)“

„url“

„month“

„search“

„week“

„range“

„time“

„number“

„color“

AUTOFOCUS

HTML-Code

```
<input type="text" autofocus>
```

Mit dem Attribut „autofocus“ haben Sie die Möglichkeit den Cursor direkt nach dem Laden einer Seite in einem Eingabefeld zu platzieren.

Das Attribut selbst erhält keinen Wert. Sie können jedoch alternativ den Wert „true“ vergeben.

PLACEHOLDER

HTML-Code

```
<input type="text" placeholder="Bitte Namen eingeben">
```

Mit Hilfe von „placeholder“ können Sie einen Text vergeben, der als Platzhalter für die eigentliche Eingabe dient. Wird der Cursor im betreffenden Eingabefeld platziert, verschwindet der Platzhalter. Dem Benutzer steht also ein leeres Feld zur Verfügung.

Verliert das Feld den Fokus wird geprüft ob ein Wert eingetragen wurde. Ist das Feld leer, wird wieder der Platzhalter verwendet.

REQUIRED

HTML-Code

```
<input type="text" required>
```

Das Attribut `required` dient der Formularüberprüfung. Enthält ein Eingabefeld das Attribut, muss es vom Besucher ausgefüllt werden. Andernfalls wird eine Fehlermeldung angezeigt und das Absenden des Formulars verhindert.

IN-PLACE-EDITING

Bisher waren Eingaben durch den Benutzer ausschließlich über Eingabefelder möglich. Mit dem Attribut „contenteditable“ können Sie nun jedoch jeden Text auf Ihrer Seite bearbeitbar machen.

Im Gegensatz zu einem Formular können Sie die Daten jedoch nicht ohne weiteres versenden oder speichern. Hier benötigen Sie eine JavaScript-Lösung.

HTML

```
<div contenteditable="true">
```

```
  Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea.
```

```
</div>
```