

Η **CSS (Cascading Style Sheets)** χρησιμοποιείται για τη διαμόρφωση της εμφάνισης μιας ιστοσελίδας. Με τη CSS καθορίζουμε χρώματα, γραμματοσειρές, διαστάσεις, αποστάσεις, περιγράμματα και τη θέση των στοιχείων.

Στη CSS χρησιμοποιούμε **κανόνες**. Κάθε κανόνας αποτελείται από έναν επιλογέα και μία ή περισσότερες δηλώσεις. Κάθε δήλωση περιλαμβάνει μια ιδιότητα και την τιμή της.

```
p {
    color: blue;
    font-size: 18px;
}
```

Στο παράδειγμα:

- Το p είναι ο **επιλογέας** και επιλέγει τις παραγράφους.
- Το color είναι η ιδιότητα που καθορίζει το χρώμα του κειμένου.
- Το blue είναι η τιμή της ιδιότητας.
- Το font-size καθορίζει το μέγεθος των γραμμάτων.
- Οι δηλώσεις γράφονται μέσα σε άγκιστρα { } και χωρίζονται με ελληνικό ερωτηματικό ;.

1. Τρόποι εφαρμογής της CSS

Η CSS μπορεί να εφαρμοστεί με τρεις βασικούς τρόπους.

Εξωτερικό αρχείο CSS

Οι κανόνες γράφονται σε ξεχωριστό αρχείο, για παράδειγμα style.css. Το αρχείο συνδέεται με την HTML μέσα στο <head>:

```
<link rel="stylesheet" href="style.css">
```

Περιεχόμενο του style.css:

```
body {
    background-color: lightgray;
}

h1 {
    color: navy;
}
```

Εσωτερική CSS

Οι κανόνες γράφονται μέσα σε ετικέτα <style> στο έγγραφο HTML:

```
<head>
    <style>
        h1 {
            color: navy;
        }

        p {
```

```

        font-size: 18px;
    }
</style>
</head>

```

CSS απευθείας σε ένα στοιχείο

Οι δηλώσεις γράφονται στο χαρακτηριστικό style:

```

<p style="color: blue; font-size: 18px;">
    Κείμενο με μπλε γράμματα.
</p>

```

Για ιστοσελίδες με πολλές σελίδες, το εξωτερικό αρχείο διευκολύνει την οργάνωση και τη συντήρηση.

2. Βασικοί επιλογείς

Οι επιλογείς καθορίζουν σε ποια στοιχεία HTML εφαρμόζονται οι κανόνες.

Επιλογέας

Χρήση

p	Επιλέγει όλα τα στοιχεία <p>.
.note	Επιλέγει στοιχεία με κλάση note.
#intro	Επιλέγει το στοιχείο με αναγνωριστικό intro.
*	Επιλέγει όλα τα στοιχεία.
h1, h2, h3	Επιλέγει όλες τις επικεφαλίδες αυτών των τριών επιπέδων.
p.note	Επιλέγει παραγράφους με κλάση note.
.card.active	Επιλέγει στοιχεία που έχουν και τις δύο κλάσεις.

Παράδειγμα HTML:

```

<h1 id="main-title">Μάθημα CSS</h1>

<p class="note">Αυτή είναι μια σημείωση.</p>

<p>Αυτή είναι μια απλή παράγραφος.</p>

```

Αντίστοιχη CSS:

```

#main-title {
    color: navy;
}

.note {
    background-color: lightyellow;
}

p {
    font-size: 18px;
}

```

Μια κλάση μπορεί να χρησιμοποιηθεί σε πολλά στοιχεία. Ένα id πρέπει να είναι μοναδικό μέσα στο ίδιο έγγραφο HTML.

3. Επιλογείς με βάση τη σχέση των στοιχείων

Επιλογέας	Χρήση
article p	Επιλέγει παραγράφους που βρίσκονται οπουδήποτε μέσα σε <article>.
article > p	Επιλέγει παραγράφους που είναι άμεσα παιδιά του <article>.
h2 + p	Επιλέγει την παράγραφο που ακολουθεί αμέσως μετά από ένα <h2>, με κοινό γονικό στοιχείο.
h2 ~ p	Επιλέγει όλες τις παραγράφους που ακολουθούν ένα <h2>, με κοινό γονικό στοιχείο.

Παράδειγμα:

```
article p {
    color: darkslategray;
}

nav > a {
    margin-right: 20px;
}

h2 + p {
    font-weight: bold;
}
```

4. Επιλογείς χαρακτηριστικών

Επιλογέας	Χρήση
[required]	Επιλέγει στοιχεία που έχουν το χαρακτηριστικό required.
[type="email"]	Επιλέγει στοιχεία με type="email".
[href^="https"]	Επιλέγει στοιχεία των οποίων το href αρχίζει με https.
[href\$=".pdf"]	Επιλέγει στοιχεία των οποίων το href τελειώνει σε .pdf.
[class*="card"]	Επιλέγει στοιχεία των οποίων το χαρακτηριστικό class περιέχει το κείμενο card.

Παράδειγμα:

```
input[type="email"] {
    background-color: lightyellow;
}

a[href$=".pdf"] {
    font-weight: bold;
}
```

5. Ψευδοκλάσεις

Οι ψευδοκλάσεις επιλέγουν στοιχεία ανάλογα με την κατάστασή τους ή τη θέση τους μέσα στο έγγραφο.

Ψευδοκλάση

Χρήση

:hover	Όταν ο δείκτης βρίσκεται πάνω στο στοιχείο.
:active	Όταν το στοιχείο ενεργοποιείται, όπως κατά το πάτημα ενός κουμπιού.
:focus	Όταν το στοιχείο έχει την εστίαση εισόδου.
:focus-visible	Όταν ο browser κρίνει ότι πρέπει να εμφανίζεται ορατή ένδειξη εστίασης.
:visited	Για συνδέσμους που έχουν ήδη επισκεφθεί οι χρήστες, με περιορισμούς στις επιτρεπόμενες αλλαγές εμφάνισης.
:first-child	Όταν το στοιχείο είναι το πρώτο παιδί του γονέα του.
:last-child	Όταν είναι το τελευταίο παιδί του γονέα του.
:nth-child(n)	Επιλέγει στοιχεία με βάση τη θέση τους μεταξύ των παιδιών.
:first-of-type	Επιλέγει το πρώτο στοιχείο του συγκεκριμένου τύπου μεταξύ αδελφών στοιχείων.
:not(...)	Εξαιρεί στοιχεία που ταιριάζουν με τον επιλογέα μέσα στις παρενθέσεις.
:checked	Επιλέγει επιλεγμένα checkbox, radio ή option.
:disabled	Επιλέγει απενεργοποιημένα χειριστήρια.
:required	Επιλέγει υποχρεωτικά πεδία φόρμας.
:valid	Επιλέγει πεδία που ικανοποιούν τους κανόνες εγκυρότητας.
:invalid	Επιλέγει πεδία που δεν ικανοποιούν τους κανόνες εγκυρότητας.
:root	Επιλέγει το στοιχείο ρίζας, συνήθως το <html>.

Παράδειγμα:

```
a:hover {
    color: red;
}

button:focus-visible {
    outline: 3px solid orange;
}

li:first-child {
    font-weight: bold;
}

tr:nth-child(even) {
    background-color: #eeeeee;
}

input:disabled {
    background-color: #dddddd;
}
```

Στο :nth-child() μπορούμε να χρησιμοποιήσουμε αριθμούς, τις λέξεις odd και even ή εκφράσεις όπως 3n.

6. Ψευδοστοιχεία

Τα ψευδοστοιχεία επιτρέπουν τη μορφοποίηση συγκεκριμένου μέρους ενός στοιχείου ή την προσθήκη διακοσμητικού περιεχομένου.

Ψευδοστοιχείο

Χρήση

::before	Δημιουργεί περιεχόμενο πριν από το κανονικό περιεχόμενο του στοιχείου.
::after	Δημιουργεί περιεχόμενο μετά από το κανονικό περιεχόμενο του στοιχείου.
::first-letter	Μορφοποιεί το πρώτο γράμμα.
::first-line	Μορφοποιεί την πρώτη γραμμή.
::selection	Μορφοποιεί το κείμενο που επιλέγει ο χρήστης.
::placeholder	Μορφοποιεί την υπόδειξη μέσα σε ένα πεδίο φόρμας.
::marker	Μορφοποιεί τον δείκτη ενός στοιχείου λίστας.

Παράδειγμα:

```
.note::before {
    content: "★ ";
    color: orange;
}

p::first-letter {
    font-size: 28px;
}

::selection {
    background-color: yellow;
    color: black;
}

input::placeholder {
    color: gray;
}
```

Οι βασικές πληροφορίες της σελίδας πρέπει να γράφονται στην HTML. Τα ::before και ::after είναι κατάλληλα κυρίως για διακοσμητικές προσθήκες.

7. Χρώματα και διαφάνεια

Ιδιότητα

Χρήση

color	Χρώμα κειμένου.
background-color	Χρώμα φόντου.
opacity	Διαφάνεια ολόκληρου του στοιχείου, μαζί με το περιεχόμενό του.

Τα χρώματα μπορούν να δηλωθούν με διάφορους τρόπους:

```
h1 {
    color: blue;
}

h2 {
    color: #006699;
}
```

```
p {
  color: rgb(40, 40, 40);
}

.note {
  background-color: rgba(255, 200, 0, 0.3);
}

footer {
  background-color: hsl(210, 50%, 20%);
}
```

Παράδειγμα διαφάνειας:

```
.photo {
  opacity: 0.7;
}
```

Η τιμή 1 σημαίνει πλήρη αδιαφάνεια και η τιμή 0 πλήρη διαφάνεια. Με ένα χρώμα που περιλαμβάνει διαφάνεια μπορούμε να επηρεάσουμε μόνο το συγκεκριμένο χρώμα, χωρίς να κάνουμε διαφανές το κείμενο.

8. Γραμματοσειρές

Ιδιότητα

Χρήση

font-family Επιλέγει γραμματοσειρά και εναλλακτικές γραμματοσειρές.

font-size Καθορίζει το μέγεθος των γραμμάτων.

font-weight Καθορίζει το βάρος των γραμμάτων.

font-style Καθορίζει κανονική ή πλάγια γραφή.

font-variant Ελέγχει τυπογραφικές παραλλαγές, όπως μικρά κεφαλαία.

line-height Καθορίζει το ύψος της γραμμής.

font Συνοπτική δήλωση πολλών ιδιοτήτων γραμματοσειράς.

Παράδειγμα:

```
body {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 18px;
  line-height: 1.6;
}

h1 {
  font-size: 36px;
  font-weight: bold;
}

.subtitle {
  font-style: italic;
}
```

Στο **font-family**, ο browser χρησιμοποιεί την πρώτη διαθέσιμη γραμματοσειρά της λίστας.

Συνοπτική γραφή:

```
p {
  font: 18px/1.6 Arial, sans-serif;
}
```

9. Μορφοποίηση και στοίχιση κειμένου

Ιδιότητα	Χρήση
text-align	Στοιχίζει το περιεχόμενο σε γραμμή: αριστερά, δεξιά, στο κέντρο ή με πλήρη στοίχιση.
text-decoration	Προσθέτει γραμμές, όπως υπογράμμιση ή διαγραφή.
text-transform	Μετατρέπει την εμφάνιση των γραμμάτων σε κεφαλαία ή πεζά.
text-indent	Καθορίζει την εσοχή της πρώτης γραμμής.
letter-spacing	Καθορίζει την απόσταση μεταξύ χαρακτήρων.
word-spacing	Καθορίζει την απόσταση μεταξύ λέξεων.
text-shadow	Προσθέτει σκιά στο κείμενο.
white-space	Καθορίζει πώς αντιμετωπίζονται τα κενά και οι αλλαγές γραμμής.
overflow-wrap	Επιτρέπει το σπάσιμο μεγάλων λέξεων όταν χρειάζεται.
word-break	Ρυθμίζει τους κανόνες διακοπής λέξεων.
text-overflow	Καθορίζει την ένδειξη για κείμενο που κόβεται λόγω υπερχειλίσσης.

Παράδειγμα:

```
h1 {
  text-align: center;
  text-transform: uppercase;
  letter-spacing: 2px;
}

p {
  text-indent: 20px;
}

.old-price {
  text-decoration: line-through;
}

.title {
  text-shadow: 2px 2px 4px gray;
}
```

Κείμενο που εμφανίζεται σε μία γραμμή και κόβεται με αποσιωπητικά:

```
.short-text {
  width: 250px;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
}
```

10. Μονάδες μέτρησης

Μονάδα	Χρήση
px	Εικονοστοιχείο CSS.
%	Ποσοστό, του οποίου η βάση εξαρτάται από την ιδιότητα.
em	Μονάδα σχετική με το μέγεθος γραμματοσειράς.
rem	Μονάδα σχετική με το μέγεθος γραμματοσειράς του στοιχείου ρίζας.
vw	Το 1% του πλάτους του viewport.
vh	Το 1% του ύψους του viewport.
ch	Μονάδα βασισμένη στο πλάτος του χαρακτήρα «0» της γραμματοσειράς.
fr	Μερίδιο του διαθέσιμου χώρου σε διάταξη Grid.
s	Δευτερόλεπτα, για διάρκεια μεταβάσεων και κινήσεων.
ms	Χιλιοστά του δευτερολέπτου.
deg	Μοίρες, για γωνίες περιστροφής και άλλες σχετικές τιμές.

Παράδειγμα:

```
h1 {
  font-size: 2rem;
}

.container {
  width: 90%;
  max-width: 1000px;
}

.hero {
  min-height: 80vh;
}
```

Αν το μέγεθος γραμματοσειράς του στοιχείου ρίζας είναι 16px, τότε το 2rem αντιστοιχεί σε 32px.

11. Διαστάσεις στοιχείων

Ιδιότητα	Χρήση
width	Καθορίζει το πλάτος.
height	Καθορίζει το ύψος.
min-width	Καθορίζει το ελάχιστο πλάτος.
max-width	Καθορίζει το μέγιστο πλάτος.
min-height	Καθορίζει το ελάχιστο ύψος.
max-height	Καθορίζει το μέγιστο ύψος.
box-sizing	Καθορίζει αν οι δηλωμένες διαστάσεις περιλαμβάνουν το padding και το border.
aspect-ratio	Καθορίζει την επιθυμητή αναλογία πλάτους προς ύψος.

Παράδειγμα:

```
* {
  box-sizing: border-box;
}

.container {
  width: 90%;
  max-width: 1100px;
}

.card {
  min-height: 200px;
}

.thumbnail {
  width: 300px;
  aspect-ratio: 16 / 9;
}
```

Με `box-sizing: border-box`, το δηλωμένο πλάτος περιλαμβάνει το περιεχόμενο, το εσωτερικό περιθώριο και το περίγραμμα.

12. Το μοντέλο κουτιού

Κάθε στοιχείο αντιμετωπίζεται ως κουτί. Το κουτί περιλαμβάνει:

- **Content:** το περιεχόμενο.
- **Padding:** την απόσταση μεταξύ περιεχομένου και περιγράμματος.
- **Border:** το περίγραμμα.
- **Margin:** την εξωτερική απόσταση από άλλα στοιχεία.

Ιδιότητα	Χρήση
<code>margin</code>	Εξωτερικά περιθώρια.
<code>margin-top</code>	Εξωτερικό περιθώριο επάνω.
<code>margin-right</code>	Εξωτερικό περιθώριο δεξιά.
<code>margin-bottom</code>	Εξωτερικό περιθώριο κάτω.
<code>margin-left</code>	Εξωτερικό περιθώριο αριστερά.
<code>padding</code>	Εσωτερικά περιθώρια.
<code>padding-top</code>	Εσωτερικό περιθώριο επάνω.
<code>padding-right</code>	Εσωτερικό περιθώριο δεξιά.
<code>padding-bottom</code>	Εσωτερικό περιθώριο κάτω.
<code>padding-left</code>	Εσωτερικό περιθώριο αριστερά.

Παράδειγμα:

```
.card {
  margin: 20px;
  padding: 15px;
  border: 1px solid gray;
}
```

Η συνοπτική γραφή ακολουθεί τους εξής κανόνες:

```
/* Όλες οι πλευρές */
margin: 20px;

/* Πάνω και κάτω — δεξιά και αριστερά */
margin: 10px 20px;

/* Πάνω — δεξιά και αριστερά — κάτω */
margin: 10px 20px 30px;

/* Πάνω — δεξιά — κάτω — αριστερά */
margin: 10px 20px 30px 40px;
```

Οι ίδιοι κανόνες ισχύουν και για το padding.

Κεντράρισμα ενός κουτιού περιορισμένου πλάτους:

```
.container {
  max-width: 1000px;
  margin: 0 auto;
}
```

13. Περιγράμματα, γωνίες και σκιές

Ιδιότητα	Χρήση
border	Καθορίζει συνολικά πάχος, τύπο και χρώμα περιγράμματος.
border-width	Πάχος περιγράμματος.
border-style	Τύπος περιγράμματος.
border-color	Χρώμα περιγράμματος.
border-top	Περίγραμμα επάνω.
border-right	Περίγραμμα δεξιά.
border-bottom	Περίγραμμα κάτω.
border-left	Περίγραμμα αριστερά.
border-radius	Στρογγυλεύει τις γωνίες.
box-shadow	Προσθέτει σκιά στο κουτί.
outline	Προσθέτει εξωτερική γραμμή που δεν καταλαμβάνει χώρο στη διάταξη.
outline-offset	Καθορίζει την απόσταση της εξωτερικής γραμμής από το στοιχείο.

Παράδειγμα:

```
.card {
  border: 2px solid #cccccc;
  border-radius: 12px;
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);
}

button:focus-visible {
  outline: 3px solid navy;
  outline-offset: 3px;
}
```

Συνηθισμένες τιμές του border-style είναι οι solid, dashed, dotted, double και none.

14. Φόντο

Ιδιότητα	Χρήση
background-color	Χρώμα φόντου.
background-image	Εικόνα ή διαβάθμιση χρωμάτων φόντου.
background-repeat	Καθορίζει αν και πώς επαναλαμβάνεται η εικόνα.
background-position	Καθορίζει τη θέση της εικόνας.
background-size	Καθορίζει το μέγεθος της εικόνας φόντου.
background-attachment	Καθορίζει τη συμπεριφορά του φόντου κατά την κύλιση.
background	Συνοπτική δήλωση ιδιοτήτων φόντου.

Παράδειγμα:

```
.hero {
  background-image: url("images/background.jpg");
  background-repeat: no-repeat;
  background-position: center;
  background-size: cover;
}
```

Στο background-size:

- Το cover καλύπτει ολόκληρη την περιοχή, με πιθανή αποκοπή μέρους της εικόνας.
- Το contain εμφανίζει ολόκληρη την εικόνα, με πιθανό κενό χώρο.

Διαβάθμιση χρωμάτων:

```
.banner {
  background-image: linear-gradient(to right, navy, skyblue);
}
```

15. Εμφάνιση και υπερχείλιση

Ιδιότητα	Χρήση
display	Καθορίζει τον τρόπο συμμετοχής του στοιχείου στη διάταξη.
visibility	Καθορίζει αν το στοιχείο είναι ορατό.
overflow	Καθορίζει τι συμβαίνει όταν το περιεχόμενο ξεπερνά τα όρια του κουτιού.
overflow-x	Ελέγχει την οριζόντια υπερχείλιση.
overflow-y	Ελέγχει την κατακόρυφη υπερχείλιση.

Συνηθισμένες τιμές του display:

Τιμή	Χρήση
block	Στη συνηθισμένη ροή δημιουργεί κουτί σε νέα γραμμή.
inline	Συμμετέχει στη γραμμή του κειμένου.

Τιμή	Χρήση
inline-block	Συμμετέχει στη γραμμή και επιτρέπει διαστάσεις κουτιού.
none	Αφαιρεί το στοιχείο από την εμφάνιση και τη διάταξη.
flex	Δημιουργεί διάταξη Flexbox για τα παιδιά του.
grid	Δημιουργεί διάταξη Grid για τα παιδιά του.

Παράδειγμα:

```
.hidden {
  display: none;
}

.invisible {
  visibility: hidden;
}

.scroll-box {
  max-height: 200px;
  overflow: auto;
}
```

Το visibility: hidden κρύβει το στοιχείο, αλλά διατηρεί τον χώρο του στη διάταξη.

16. Τοποθέτηση στοιχείων

Ιδιότητα	Χρήση
position	Καθορίζει τον τρόπο τοποθέτησης.
top	Απόσταση από την επάνω πλευρά αναφοράς.
right	Απόσταση από τη δεξιά πλευρά αναφοράς.
bottom	Απόσταση από την κάτω πλευρά αναφοράς.
left	Απόσταση από την αριστερή πλευρά αναφοράς.
inset	Συνοπτική δήλωση των τεσσάρων παραπάνω αποστάσεων.
z-index	Ελέγχει τη σειρά επικάλυψης σε στοιχεία όπου εφαρμόζεται.

Τιμές του position:

Τιμή	Χρήση
static	Κανονική τοποθέτηση στη ροή. Οι top, right, bottom, left δεν το μετακινούν.
relative	Επιτρέπει μετατόπιση από την κανονική θέση, διατηρώντας τον αρχικό χώρο.
absolute	Τοποθετεί το στοιχείο έξω από την κανονική ροή, σε σχέση με το κατάλληλο πλαίσιο αναφοράς.
fixed	Τοποθετεί το στοιχείο έξω από τη ροή, συνήθως σε σχέση με το viewport.
sticky	Διατηρεί το στοιχείο στη ροή και το συγκρατεί σε καθορισμένη θέση κατά την κύλιση, μέσα στα όρια του περιέκτη του.

Παράδειγμα:

```
.card {
```

```

    position: relative;
}

.badge {
    position: absolute;
    top: 10px;
    right: 10px;
}

.menu {
    position: sticky;
    top: 0;
    z-index: 10;
}

```

Το z-index λειτουργεί μέσα σε συμφραζόμενα στοίβαξης. Ένας πολύ μεγάλος αριθμός δεν εξασφαλίζει ότι το στοιχείο θα εμφανιστεί πάνω από κάθε άλλο στοιχείο της σελίδας.

17. Flexbox

Το Flexbox οργανώνει στοιχεία κυρίως κατά μήκος ενός άξονα, σε σειρά ή στήλη.

Ιδιότητα	Χρήση
display: flex	Ενεργοποιεί το Flexbox στον περιέκτη.
flex-direction	Καθορίζει την κατεύθυνση: σειρά ή στήλη.
flex-wrap	Επιτρέπει τη δημιουργία περισσότερων γραμμών.
flex-flow	Συνδυάζει flex-direction και flex-wrap.
justify-content	Κατανέμει τον χώρο στον κύριο άξονα.
align-items	Στοιχίζει τα στοιχεία στον εγκάρσιο άξονα.
align-content	Κατανέμει τις γραμμές στον εγκάρσιο άξονα όταν υπάρχουν πολλές γραμμές και διαθέσιμος χώρος.
gap	Καθορίζει την απόσταση μεταξύ στοιχείων.
row-gap	Καθορίζει την απόσταση μεταξύ γραμμών.
column-gap	Καθορίζει την απόσταση μεταξύ στηλών.
flex-grow	Καθορίζει πώς ένα στοιχείο συμμετέχει στην κατανομή του επιπλέον χώρου.
flex-shrink	Καθορίζει πώς ένα στοιχείο μπορεί να συρρικνωθεί όταν ο χώρος δεν επαρκεί.
flex-basis	Καθορίζει το αρχικό μέγεθος στον κύριο άξονα.
flex	Συνοπτική δήλωση των flex-grow, flex-shrink και flex-basis.
align-self	Ορίζει διαφορετική στοίχιση για ένα συγκεκριμένο στοιχείο.
order	Αλλάζει την οπτική σειρά ενός στοιχείου.

Παράδειγμα HTML:

```

<div class="cards">
  <div class="card">Πρώτο μάθημα</div>
  <div class="card">Δεύτερο μάθημα</div>
  <div class="card">Τρίτο μάθημα</div>

```

```
</div>
```

CSS:

```
.cards {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
}

.card {
  flex: 1 1 250px;
  padding: 20px;
  background-color: lightblue;
}
```

Κεντράρισμα περιεχομένου σε δύο άξονες:

```
.container {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 300px;
}
```

Με flex-direction: row, ο κύριος άξονας ακολουθεί τη σειρά. Με flex-direction: column, ακολουθεί τη στήλη.

18. Grid

Το Grid οργανώνει στοιχεία σε γραμμές και στήλες.

Ιδιότητα	Χρήση
display: grid	Ενεργοποιεί το Grid.
grid-template-columns	Καθορίζει τις στήλες.
grid-template-rows	Καθορίζει τις γραμμές.
grid-template-areas	Περιγράφει τη διάταξη με ονομασμένες περιοχές.
grid-column	Καθορίζει σε ποιες στήλες τοποθετείται ένα στοιχείο.
grid-row	Καθορίζει σε ποιες γραμμές τοποθετείται ένα στοιχείο.
grid-area	Αναθέτει ένα στοιχείο σε περιοχή ή καθορίζει τη θέση του.
grid-auto-columns	Καθορίζει το μέγεθος στηλών που δημιουργούνται αυτόματα.
grid-auto-rows	Καθορίζει το μέγεθος γραμμών που δημιουργούνται αυτόματα.
grid-auto-flow	Ελέγχει την αυτόματη τοποθέτηση των στοιχείων.
justify-items	Στοιχίζει τα στοιχεία μέσα στις περιοχές τους στον άξονα της γραμμής κειμένου.
align-items	Στοιχίζει τα στοιχεία στον άλλο άξονα.
place-items	Συνδυάζει align-items και justify-items.
gap	Καθορίζει τις αποστάσεις μεταξύ γραμμών και στηλών.

Παράδειγμα τριών ίσων στηλών:

```
.cards {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 20px;
}
```

Παράδειγμα διάταξης με περιοχές:

```
.layout {
  display: grid;
  grid-template-columns: 200px 1fr;
  grid-template-areas:
    "header header"
    "sidebar main"
    "footer footer";
  gap: 20px;
}

header {
  grid-area: header;
}

aside {
  grid-area: sidebar;
}

main {
  grid-area: main;
}

footer {
  grid-area: footer;
}
```

Η συνάρτηση repeat() επαναλαμβάνει έναν ορισμό. Η τιμή 1fr αντιστοιχεί σε ένα μερίδιο του διαθέσιμου χώρου.

19. Εικόνες και πολυμέσα

Ιδιότητα

Χρήση

object-fit Καθορίζει πώς μια εικόνα ή ένα βίντεο προσαρμόζεται μέσα στο κουτί του.

object-position Καθορίζει τη θέση του περιεχομένου μέσα στο κουτί.

filter Εφαρμόζει οπτικά εφέ, όπως θόλωση ή αλλαγή φωτεινότητας.

Εικόνα που προσαρμόζεται στο διαθέσιμο πλάτος:

```
img {
  max-width: 100%;
  height: auto;
}
```

Εικόνες με κοινές διαστάσεις:

```
.thumbnail {
  width: 300px;
  height: 200px;
  object-fit: cover;
  object-position: center;
}
```

Οπτικά εφέ:

```
.grayscale {
  filter: grayscale(100%);
}

.blurred {
  filter: blur(3px);
}

.bright {
  filter: brightness(120%);
}
```

20. Λίστες και πίνακες

Ιδιότητα	Χρήση
list-style-type	Καθορίζει τον τύπο δείκτη της λίστας.
list-style-position	Καθορίζει αν ο δείκτης βρίσκεται μέσα ή έξω από το κουτί περιεχομένου.
list-style-image	Χρησιμοποιεί εικόνα ως δείκτη.
list-style	Συνοπτική δήλωση ιδιοτήτων λίστας.
border-collapse	Καθορίζει αν τα περιγράμματα των κελιών συγχωνεύονται.
border-spacing	Καθορίζει την απόσταση μεταξύ κελιών όταν τα περιγράμματα είναι χωριστά.
table-layout	Καθορίζει τον αλγόριθμο διάταξης του πίνακα.
caption-side	Καθορίζει την πλευρά εμφάνισης του τίτλου του πίνακα.

Παράδειγμα:

```
ul {
  list-style-type: square;
}

nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
}

table {
  width: 100%;
  border-collapse: collapse;
}

th,
td {
```

```
border: 1px solid gray;
padding: 10px;
text-align: left;
}
```

21. Κουμπιά, φόρμες και αλληλεπίδραση

Ιδιότητα	Χρήση
cursor	Καθορίζει την εμφάνιση του δείκτη.
resize	Ελέγχει αν ο χρήστης μπορεί να αλλάξει τις διαστάσεις ενός στοιχείου που το υποστηρίζει.
accent-color	Καθορίζει το χρώμα έμφασης σε ορισμένα χειριστήρια φόρμας.
appearance	Ελέγχει την εγγενή εμφάνιση χειριστηρίων.
pointer-events	Ελέγχει αν ένα στοιχείο μπορεί να γίνει στόχος ενεργειών του δείκτη.
user-select	Ελέγχει τη δυνατότητα επιλογής κειμένου.
caret-color	Καθορίζει το χρώμα του δρομέα εισαγωγής κειμένου.

Παράδειγμα:

```
input,
textarea,
select {
    font: inherit;
    padding: 10px;
    border: 1px solid gray;
    border-radius: 6px;
}

textarea {
    resize: vertical;
}

input[type="checkbox"] {
    accent-color: navy;
}

button {
    font: inherit;
    background-color: navy;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 6px;
    cursor: pointer;
}

button:hover {
    background-color: royalblue;
}
```

Το `pointer-events: none` δεν αποτελεί πλήρη απενεργοποίηση ενός κουμπιού. Για απενεργοποίηση χειριστηρίου χρησιμοποιείται το κατάλληλο χαρακτηριστικό HTML, όπως `disabled`.

22. Μετασχηματισμοί

Ιδιότητα ή συνάρτηση	Χρήση
transform	Εφαρμόζει μετασχηματισμούς στο στοιχείο.
translate()	Μετακινεί το στοιχείο.
rotate()	Περιστρέφει το στοιχείο.
scale()	Μεγαλώνει ή μικραίνει το στοιχείο.
skew()	Παραμορφώνει το στοιχείο υπό γωνία.
transform-origin	Καθορίζει το σημείο αναφοράς του μετασχηματισμού.

Παραδείγματα:

```
.move {
  transform: translate(20px, 10px);
}

.rotate {
  transform: rotate(15deg);
}

.enlarge {
  transform: scale(1.2);
}

.combined {
  transform: translateY(10px) rotate(5deg);
}
```

Οι μετασχηματισμοί αλλάζουν την οπτική εμφάνιση χωρίς να αναδιατάσσουν τον χώρο που έχει ήδη δεσμευτεί για το στοιχείο στην κανονική ροή. Η σειρά των μετασχηματισμών επηρεάζει το αποτέλεσμα.

23. Μεταβάσεις

Οι μεταβάσεις κάνουν την αλλαγή μιας ιδιότητας να πραγματοποιείται σταδιακά.

Ιδιότητα	Χρήση
transition-property	Καθορίζει ποια ιδιότητα θα αλλάξει σταδιακά.
transition-duration	Καθορίζει τη διάρκεια.
transition-timing-function	Καθορίζει τον ρυθμό της αλλαγής.
transition-delay	Καθορίζει την καθυστέρηση πριν από την αλλαγή.
transition	Συνοπτική δήλωση των παραπάνω.

Παράδειγμα:

```
button {
  background-color: navy;
  transform: scale(1);
  transition:
    background-color 0.3s ease,
```

```

        transform 0.3s ease;
    }

    button:hover {
        background-color: royalblue;
        transform: scale(1.05);
    }

```

Συνηθισμένες τιμές χρονισμού είναι οι linear, ease, ease-in, ease-out και ease-in-out.

24. Κινήσεις με @keyframes

Με το @keyframes καθορίζουμε στάδια μιας κίνησης.

Ιδιότητα	Χρήση
animation-name	Όνομα της κίνησης.
animation-duration	Διάρκεια ενός κύκλου.
animation-timing-function	Ρυθμός της κίνησης.
animation-delay	Καθυστέρηση έναρξης.
animation-iteration-count	Αριθμός επαναλήψεων.
animation-direction	Κατεύθυνση εκτέλεσης.
animation-fill-mode	Καθορίζει την εφαρμογή των τιμών πριν ή μετά την ενεργή διάρκεια.
animation-play-state	Καθορίζει αν η κίνηση εκτελείται ή είναι σε παύση.
animation	Συνοπτική δήλωση ιδιοτήτων κίνησης.

Παράδειγμα:

```

@keyframes move {
    from {
        transform: translateX(0);
    }

    to {
        transform: translateX(100px);
    }
}

.box {
    animation: move 2s ease-in-out infinite alternate;
}

```

Το infinite ορίζει συνεχείς επαναλήψεις. Το alternate εναλλάσσει την κατεύθυνση της κίνησης.

25. Προσαρμογή σε διαφορετικές οθόνες

Με το @media εφαρμόζουμε κανόνες όταν ικανοποιούνται συγκεκριμένες συνθήκες.

```

.cards {

```

```

    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 20px;
}

@media (max-width: 800px) {
    .cards {
        grid-template-columns: repeat(2, 1fr);
    }
}

@media (max-width: 500px) {
    .cards {
        grid-template-columns: 1fr;
    }
}

```

Στο παράδειγμα, οι κάρτες εμφανίζονται σε τρεις, δύο ή μία στήλη, ανάλογα με το πλάτος του viewport.

Προσαρμογή για εκτύπωση:

```

@media print {
    nav,
    button {
        display: none;
    }

    body {
        color: black;
        background-color: white;
    }
}

```

Προσαρμογή όταν ο χρήστης προτιμά μειωμένη κίνηση:

```

@media (prefers-reduced-motion: reduce) {
    .box {
        animation: none;
    }
}

```

26. Μεταβλητές CSS

Οι προσαρμοσμένες ιδιότητες, που συχνά ονομάζονται μεταβλητές CSS, αποθηκεύουν τιμές για επαναχρησιμοποίηση.

```

:root {
    --main-color: #006699;
    --background-color: #f5f5f5;
    --spacing: 20px;
}

body {
    background-color: var(--background-color);
}

h1 {

```

```

    color: var(--main-color);
}

.card {
  padding: var(--spacing);
  border: 1px solid var(--main-color);
}

```

Τα ονόματά τους αρχίζουν με δύο παύλες --. Η συνάρτηση `var()` χρησιμοποιεί την αποθηκευμένη τιμή.

Μπορούμε να ορίσουμε και εναλλακτική τιμή:

```

p {
  color: var(--text-color, black);
}

```

27. Χρήσιμες συναρτήσεις

Συνάρτηση	Χρήση
<code>calc()</code>	Υπολογίζει τιμή με αριθμητικές πράξεις.
<code>min()</code>	Επιλέγει τη μικρότερη από τις δοσμένες τιμές.
<code>max()</code>	Επιλέγει τη μεγαλύτερη από τις δοσμένες τιμές.
<code>clamp()</code>	Περιορίζει μια προτιμώμενη τιμή μεταξύ ελάχιστου και μέγιστου ορίου.
<code>var()</code>	Χρησιμοποιεί προσαρμοσμένη ιδιότητα.
<code>url()</code>	Δηλώνει τη διεύθυνση ενός πόρου.
<code>repeat()</code>	Επαναλαμβάνει ορισμούς γραμμών ή στηλών στο Grid.
<code>minmax()</code>	Ορίζει ελάχιστο και μέγιστο μέγεθος μιας γραμμής ή στήλης Grid.

Παραδείγματα:

```

.container {
  width: calc(100% - 40px);
}

.article {
  width: min(90%, 900px);
}

h1 {
  font-size: clamp(24px, 4vw, 48px);
}

.cards {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
  gap: 20px;
}

```

Το `clamp(24px, 4vw, 48px)` επιτρέπει στο μέγεθος να μεταβάλλεται με το πλάτος της οθόνης, διατηρώντας τα όρια των 24px και 48px.

28. Πρόσθετοι κανόνες με @

Κανόνας	Χρήση
@media	Εφαρμόζει CSS υπό συγκεκριμένες συνθήκες μέσου ή viewport.
@keyframes	Ορίζει στάδια κίνησης.
@font-face	Δηλώνει μια γραμματοσειρά που φορτώνεται από αρχείο.
@import	Εισάγει κανόνες από άλλο αρχείο CSS.
@supports	Εφαρμόζει κανόνες όταν ο browser υποστηρίζει μια συγκεκριμένη δυνατότητα.
@page	Καθορίζει παραμέτρους σελίδας για εκτύπωση.

Παράδειγμα γραμματοσειράς:

```
@font-face {
  font-family: "MyFont";
  src: url("fonts/myfont.woff2") format("woff2");
  font-display: swap;
}

body {
  font-family: "MyFont", Arial, sans-serif;
}
```

Έλεγχος υποστήριξης:

```
@supports (display: grid) {
  .layout {
    display: grid;
    grid-template-columns: 1fr 1fr;
  }
}
```

Ρυθμίσεις εκτύπωσης:

```
@page {
  size: A4;
  margin: 2cm;
}
```

29. Κληρονομικότητα και προτεραιότητα κανόνων

Ορισμένες ιδιότητες, όπως το χρώμα και η γραμματοσειρά, συνήθως κληρονομούνται από τον γονέα στα παιδιά του.

```
body {
  color: darkslategray;
  font-family: Arial, sans-serif;
}
```

Οι παράγραφοι και άλλα στοιχεία χρησιμοποιούν αυτές τις τιμές, εκτός αν υπάρχει κανόνας που τις αλλάζει.

Όταν περισσότεροι κανόνες επηρεάζουν την ίδια ιδιότητα, η CSS χρησιμοποιεί τον μηχανισμό της **διαδοχής (cascade)** για να επιλέξει την τελική τιμή.

Στη συνηθισμένη περίπτωση κανόνων της ίδιας προέλευσης, σημασίας και επιπέδου διαδοχής, εξετάζεται η ειδικότητα του επιλογέα:

Είδος επιλογέα	Σχετική ειδικότητα
#intro	Μεγαλύτερη από επιλογείς κλάσεων και ετικετών.
.note, [required], :hover	Μεγαλύτερη από επιλογείς ετικετών.
p, h1, ::before	Μικρότερη από επιλογείς κλάσεων και αναγνωριστικών.
*	Δεν προσθέτει ειδικότητα.

Παράδειγμα:

```
p {
  color: black;
}

.note {
  color: blue;
}
```

Η παράγραφος <p class="note"> εμφανίζεται μπλε.

Όταν δύο κανόνες έχουν ίση προτεραιότητα και ειδικότητα, υπερισχύει εκείνος που γράφεται αργότερα:

```
p {
  color: blue;
}

p {
  color: green;
}
```

Εδώ οι παράγραφοι εμφανίζονται πράσινες.

Το !important αλλάζει την προτεραιότητα μιας δήλωσης:

```
p {
  color: red !important;
}
```

Χρησιμοποιείται περιορισμένα, επειδή δυσκολεύει τις μελλοντικές αλλαγές.

30. Ειδικές τιμές ιδιοτήτων

Τιμή	Χρήση
inherit	Χρησιμοποιεί την υπολογισμένη τιμή του γονικού στοιχείου.
initial	Χρησιμοποιεί την αρχική τιμή που ορίζει η CSS για την ιδιότητα.
unset	Λειτουργεί ως inherit για κληρονομούμενες ιδιότητες και ως initial για τις υπόλοιπες.

Τιμή**Χρήση**

revert Αναίρει τη συμβολή του τρέχοντος επιπέδου προέλευσης της διαδοχής για την ιδιότητα.

Παράδειγμα:

```
button {
    font-family: inherit;
}

.special-text {
    color: initial;
}
```

Η αρχική τιμή μιας ιδιότητας δεν είναι απαραίτητα ίδια με την προεπιλεγμένη εμφάνιση που δίνει ο browser σε μια συγκεκριμένη ετικέτα.

31. Σχόλια στη CSS

Τα σχόλια γράφονται ανάμεσα σε /* και */.

```
/* Βασική εμφάνιση σελίδας */
body {
    font-family: Arial, sans-serif;
    background-color: #f5f5f5;
}

/* Κύριος τίτλος */
h1 {
    color: navy;
}
```

Μπορούν να καταλαμβάνουν περισσότερες γραμμές:

```
/*
    Κανόνες για το κύριο μενού
    της ιστοσελίδας.
*/
```

32. Ολοκληρωμένο παράδειγμα

Αρχείο HTML:

```
<!DOCTYPE html>
<html lang="el">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μάθημα CSS</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Μαθήματα σχεδιασμού ιστοσελίδων</h1>
    </header>
```

```

<main class="container">
  <section class="cards">
    <article class="card">
      <h2>HTML</h2>
      <p>Οργάνωση του περιεχομένου μιας ιστοσελίδας.</p>
      <a href="html.html">Περισσότερα</a>
    </article>

    <article class="card">
      <h2>CSS</h2>
      <p>Διαμόρφωση της εμφάνισης μιας ιστοσελίδας.</p>
      <a href="css.html">Περισσότερα</a>
    </article>
  </section>
</main>

<footer>
  <p>Σημειώσεις μαθήματος</p>
</footer>
</body>
</html>

```

```

* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: Arial, sans-serif;
  line-height: 1.6;
  color: #333333;
  background-color: #f2f2f2;
}

header {
  padding: 30px 20px;
  text-align: center;
  color: white;
  background-color: #006699;
}

.container {
  width: 90%;
  max-width: 1000px;
  margin: 30px auto;
}

.cards {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 20px;
}

.card {
  padding: 20px;
  background-color: white;
  border: 1px solid #dddddd;
  border-radius: 10px;
}

```

```

        box-shadow: 0 3px 10px rgba(0, 0, 0, 0.1);
    }

    .card h2 {
        margin-top: 0;
        color: #006699;
    }

    .card a {
        display: inline-block;
        padding: 8px 16px;
        color: white;
        background-color: #006699;
        border-radius: 5px;
        text-decoration: none;
        transition: background-color 0.3s ease;
    }

    .card a:hover {
        background-color: #004466;
    }

    .card a:focus-visible {
        outline: 3px solid #ff9900;
        outline-offset: 3px;
    }

    footer {
        padding: 15px;
        text-align: center;
        background-color: #dddddd;
    }

    @media (max-width: 600px) {
        .cards {
            grid-template-columns: 1fr;
        }
    }

```