FLEX 핵심 이론

HTML+CSS+FLEX+JQUERY 핵심 이론 가이드북(Guide Book)

코딩웍스 CSS3 강좌 – 커리큘럼

코딩웍스 유튜브 채널에서 HTML+CSS 필수 핵심이론 동영상을 보실 때 이론 PDF 교재를 참고하면서 학습하시기 바랍니다.

CSS 핵심이론 커리큘럼

Part 4 : CSS Flex(Flexible Box) 완벽 가이드

1. 부모요소에 쓰는 속성
   - display: flex / justify-content / align-items / flex-direction
   - flex-wrap / align-content
2. 자식요소에 쓰는 속성
   - flex-grow / flex-shrink / flex-basis / flex / order / align-self
   - 자식요소 사용하는 마진(margin) 속성
3. Flex 적용 예제
   - 레이아웃(1) : 수직중앙 수평중앙 중첩해서 사용하기
   - 레이아웃(2) - 부모자식으로 중첩된 div를 가로 배치하기
   - 레이아웃(3) – 상단 네비게이션 만들기
   - 레이아웃(4) – 시멘틱 태그 반응형 레이아웃

코딩웍스 CSS3 flexbox

CSS flexbox(Flexible Box) 개념과 사용하는 이유

Flexbox는 효율적인 CSS3 레이아웃 제작 방식이지만 IE와 Edge 브라우저에서 부분적으로만 지원하기 때문에 사용을 결정할 경우 신중하게 선택하시기 바랍니다.

Flexbox는 기존의 float, position, display 속성을 사용해서 HTML 요소의 배치, 정렬, 방향, 순서, 크기를 조절하는 대신에 좀 더 쉽고 효율적으로 조절할 수 있는 방법을 지원하는 CSS3 레이아웃 제작 방식입니다. 특히 반응형 레이아웃을 만들기 쉽습니다.

▼ 기존 방식으로 수직중앙 수평중앙 배치하기

```
body {
    background-color: greenyellow;  height: 100vh;
}
div {
    width: 100px;  height: 100px;
    background-color: #000;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}
```

HTML

```
<div></div>
```

▼ 플렉스 방식으로 수직중앙 수평중앙 배치하기

```
body {
    background-color: greenyellow; height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
}
div {
    width: 100px;  height: 100px;
    background-color: #000;
}
```

코딩웍스 CSS3 flexbox

부모 요소와 자식 요소에 정의하는 flexbox 속성 정리

부모요소에 쓰는 속성 (.parent)

- display: flex
- justify-content
- align-items
- flex-direction
- flex-wrap
- align-content

자식요소에 쓰는 속성 (.child)

- flex
- order
- align-self

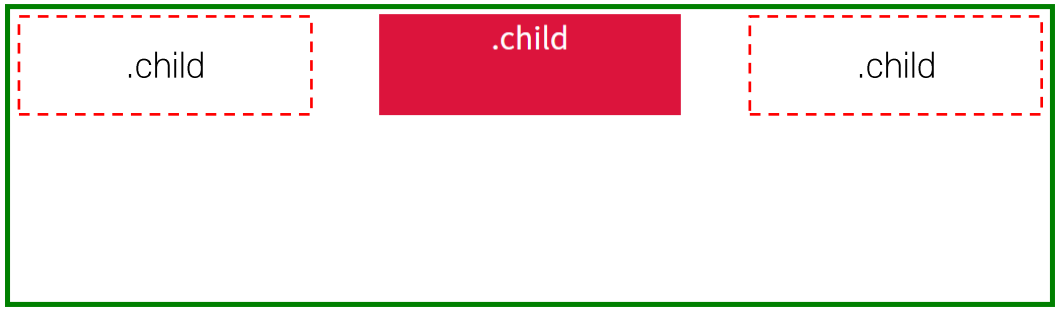
- **display**: flex | inline-flex
- **flex-direction**: row | column
- **flex-wrap**: wrap | nowrap | wrap-reverse
- **justify-content**: flex-start | flex-end | center | space-between | space-around
- **align-items**: flex-start | flex-end | center | baseline | stretch
- **align-content**: flex-start | flex-end | center | space-between | space-around | stretch

- **flex**: 숫자(정수)
- **order**: 숫자(정수)
- **align-self**: auto | flex-start | flex-end | center | baseline | stretch

코딩웍스 CSS3 flexbox - 부모요소 속성

01) 부모요소에 사용하는 Flexbox 속성 : justify-content: center | flex-start | flex-end

자식요소들의 좌측, 중앙, 우측 등 수평배치 변경합니다.



▼ 자식요소 1개를 배치하는 경우

HTML

```
<div class="parent">
  <div class="child">.child</div>
</div>
```

```
.parent {
    border: 5px solid green;  height: 500px;
    display: flex;
    justify-content: center;
}
.parent div {
    width: 300px;  color: #fff;  text-align: center;  font-size: 30px;
    margin: 5px;  background-color: crimson;
}
.child { height: 100px; }
```

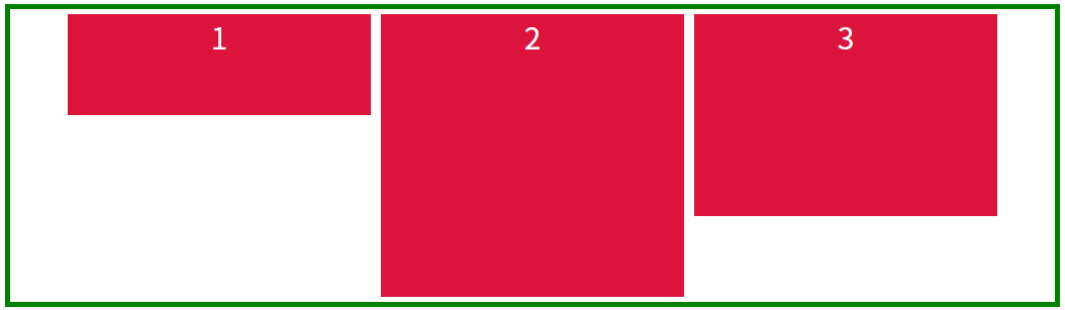
플렉스 박스를 시작할 때 무조건 사용해야 하는 속성

왼쪽 정렬 justify-content: flex-start;
우측 정렬 justify-content: flex-end;

높이 값이 없는 경우 부모요소의 높이 값에 맞춰 100%로 채워집니다.

코딩웍스 CSS3 flexbox - 부모요소 속성

01) 부모요소에 사용하는 Flexbox 속성 : justify-content: center;



▼ 자식요소 3개를 배치하는 경우

HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
</div>
```

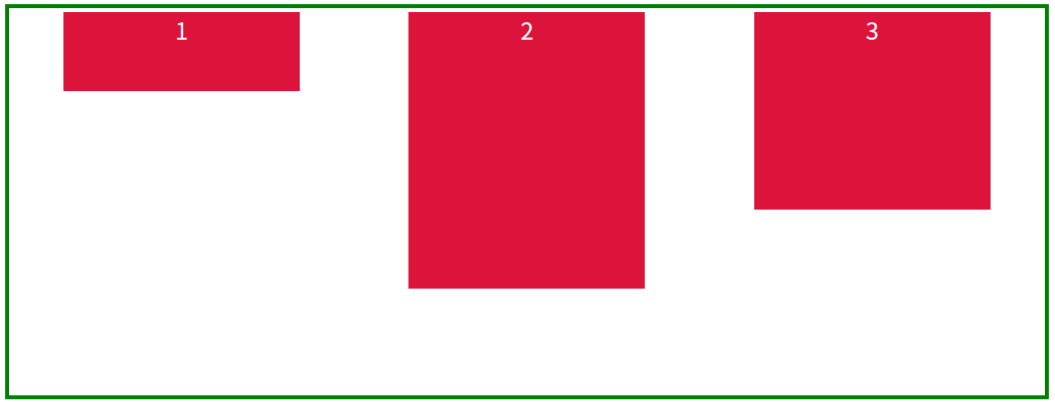
```
.parent {
    border: 5px solid green;  height: 300px;
    display: flex;
    justify-content: center;
}
.parent div {
    width: 300px;  color: #fff;
    text-align: center;
    font-size: 30px;
    margin: 5px;
    background-color: crimson;
}
.child1 { height: 100px; }
.child2 { }
.child3 { height: 200px; }
```

여러 개의 자식요소가 좌우 간격이 없이 수평 중앙 정렬

높이 값이 지정되지 않는 경우 자동으로 높이는 100%

코딩웍스 CSS3 flexbox - 부모요소 속성

01) 부모요소에 사용하는 Flexbox 속성 : justify-content: space-around;



▼ 자식요소 3개를 배치하는 경우

HTML

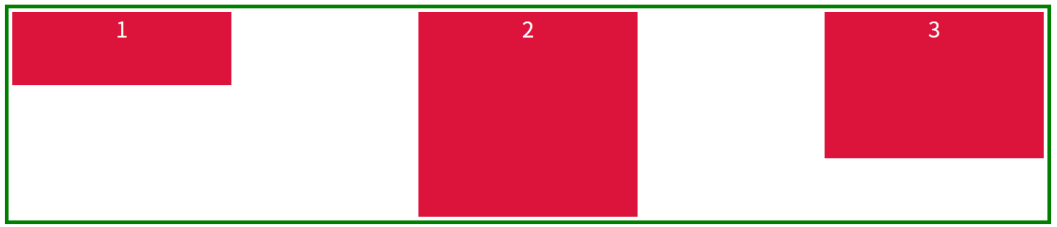
```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
</div>
```

```
.parent {
    border: 5px solid green;  height: 300px;
    display: flex;
    justify-content: space-around;
}
.parent div {
    width: 300px;  color: #fff;
    text-align: center;
    font-size: 30px;
    margin: 5px;
    background-color: crimson;
}
.child1 { height: 100px; }
.child2 { }
.child3 { height: 200px; }
```

여러 개의 자식요소끼리 좌우 간격을 일정하게 배분해서 수평 중앙에 정렬

코딩웍스 CSS3 flexbox - 부모요소 속성

01) 부모요소에 사용하는 Flexbox 속성 : justify-content: space-between;




```
.parent {
    border: 5px solid green;  height: 300px;
    display: flex;
    justify-content: space-between;
}
.parent div {
    width: 300px;  color: #fff;
    text-align: center;
    font-size: 30px;
    margin: 5px;
    background-color: crimson;
}
.child1 { height: 100px; }
.child2 { }
.child3 { height: 200px; }
```

여러 개의 자식요소를 부모요소의 좌우 여백없이 꽉 채우면서 중앙에 배치

▼ 자식요소 3개를 배치하는 경우

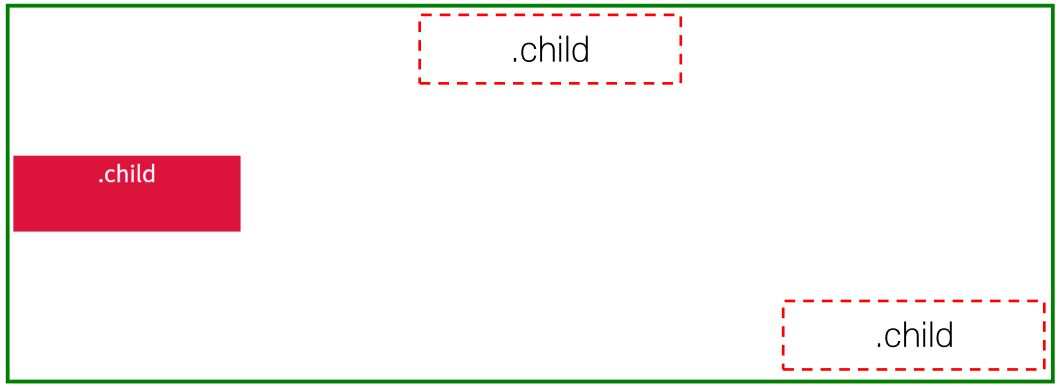
HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
</div>
```

코딩웍스 CSS3 flexbox - 부모요소 속성

02) 부모요소에 사용하는 Flexbox 속성 : align-items: center | flex-start | flex-end

자식요소들의 상단, 중앙, 하단 등 수직배치 변경합니다.



▼ 자식요소 1개를 배치하는 경우

HTML

```
<div class="parent">
  <div class="child">.child</div>
</div>
```

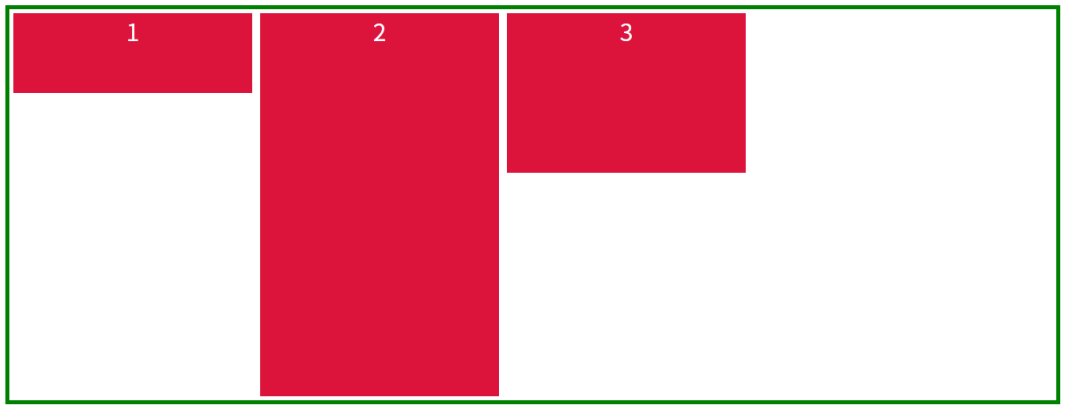
```
.parent {
    border: 5px solid green;  height: 500px;
    display: flex;
    align-items: center;
}
.parent div {
    width: 300px;  color: #fff;  text-align: center;  font-size: 30px;
    margin: 5px;  background-color: crimson;
}
.child { height: 100px; }
```

상단 정렬 align-items: flex-start;
하단 정렬 align-items: flex-end;

높이 값이 없는 경우 자식요소의 내용의 높이 만큼으로 맞춰짐.

코딩웍스 CSS3 flexbox - 부모요소 속성

02) 부모요소에 사용하는 Flexbox 속성 : align-items: stretch;



▼ 자식요소 3개를 배치하는 경우

HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
</div>
```

```
.parent {
    border: 5px solid green;  height: 300px;
    display: flex;
    align-items: stretch;
}
.parent div {
    width: 300px;  color: #fff;
    text-align: center;
    font-size: 30px;
    margin: 5px;
    background-color: crimson;
}
.child1 { height: 100px; }
.child2 { }
.child3 { height: 200px; }
```

자식요소에 높이 값이 없으면 부모요소의 높이 값에 맞게 자동으로 100% 가득 채움

코딩웍스 CSS3 flexbox - 부모요소 속성

Flexbox를 사용해서 자식요소를 부모요소의 수평 중앙 수직 중앙 배치하기

HTML

```
<div class="parent">
    <div>.child</div>
</div>
```

```
.parent {
    border: 5px solid green;
    height: 500px;
    display: flex;
    justify-content: center;
    align-items: center;
}
.parent div {
    color: #fff;
    width: 300px; height: 100px;
    text-align: center;
    font-size: 30px;
    background-color: crimson;
}
```

justify-content: center; → 자식요소를 부모요소의 수평 중앙 정렬

align-items: center; → 자식요소를 부모요소의 수직 중앙 정렬

포지션 속성 사용해서 부모요소 수평 중앙 수직 중앙

```
.parent { position: relative; }
.parent div {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}
```

코딩웍스 CSS3 flexbox - 부모요소 속성

03) 부모요소에 사용하는 Flexbox 속성 : flex-direction: row | row-reverse

자식요소의 가로배치 세로배치, 정방향 배치, 역방향 배치

HTML

```
<div class="parent">
    <div class="child1">child1</div>
    <div class="child2">child2</div>
    <div class="child3">child3</div>
</div>
```

```
.parent {
    border: 5px solid green;  height: 500px;
    display: flex;
    flex-direction: row;  /* 기본 값 */
    flex-direction: row-reverse;
}
.parent div {
    width: 300px; height: 200px;  color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;  background-color: crimson;
}
.child1 { background-color: crimson; }
.child2 { background-color: dodgerblue; }
.child3 { background-color: yellowgreen; }
```

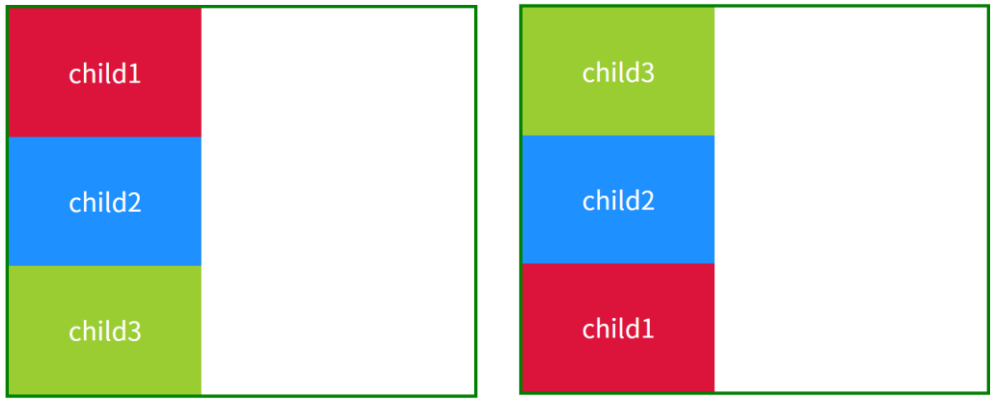
정방향 배치 가로배치

역방향 배치 가로배치

코딩웍스 CSS3 flexbox - 부모요소 속성

03) 부모요소에 사용하는 Flexbox 속성 : flex-direction: column | column-reverse

자식요소의 가로배치 세로배치, 정방향 배치, 역방향 배치



HTML

```
<div class="parent">
    <div class="child1">child1</div>
    <div class="child2">child2</div>
    <div class="child3">child3</div>
</div>
```

```
.parent {
    border: 5px solid green;  height: 500px;
    display: flex;
    flex-direction: column;  /* 기본 값 */
    flex-direction: column-reverse;
}
.parent div {
    width: 300px; height: 200px;  color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;  background-color: crimson;
}
.child1 { background-color: crimson; }
.child2 { background-color: dodgerblue; }
.child3 { background-color: yellowgreen; }
```

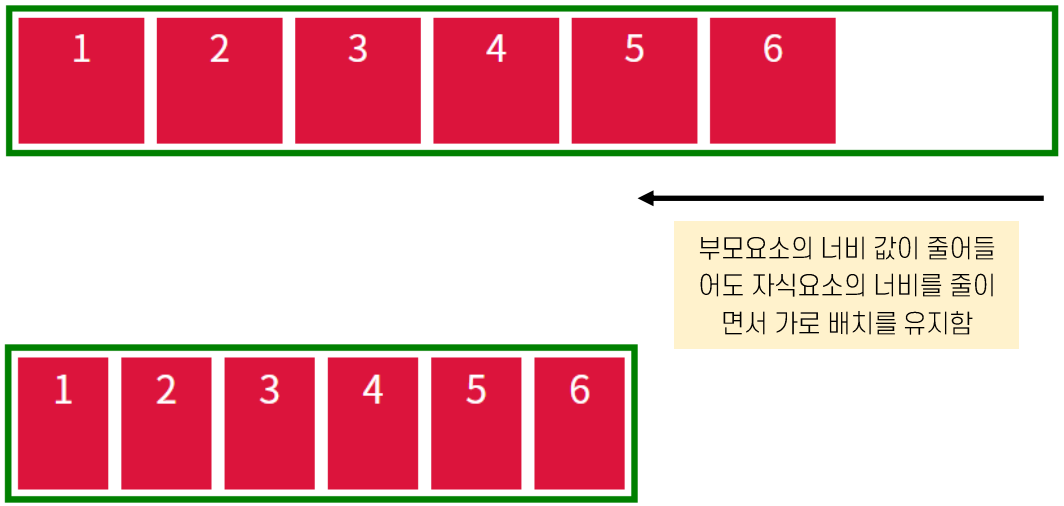
정방향 배치 세로배치

역방향 배치 세로배치

코딩웍스 CSS3 flexbox - 부모요소 속성

04) 부모요소에 사용하는 Flexbox 속성 : flex-wrap: nowrap | wrap | wrap-reverse

부모요소의 너비값 안에서 줄바꿈 없이 배치할지 줄을 바꿀지 결정




```
.parent {
    border: 5px solid green;
    display: flex;
    flex-wrap: nowrap;  /* 기본 값 */
}
```

```
.parent div {
    width: 100px;  height: 100px;
    color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;
    background-color: crimson;
}
```

HTML

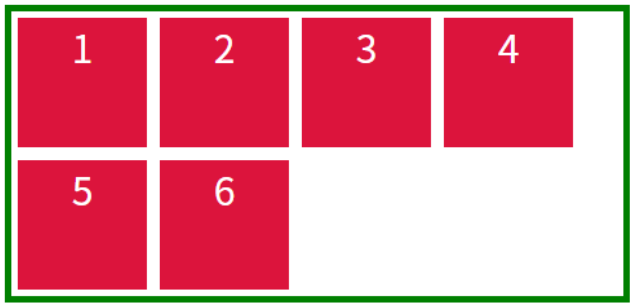
```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
    <div class="child3">4</div>
    <div class="child3">5</div>
    <div class="child3">6</div>
</div>
```

코딩웍스 CSS3 flexbox - 부모요소 속성

04) 부모요소에 사용하는 Flexbox 속성 : flex-wrap: nowrap | wrap | wrap-reverse

부모요소의 너비 값 안에서 줄 바꿈 없이 배치할지 줄을 바꿀지 결정

부모요소의 너비 값이 줄어들면 자식요소의 너비를 줄이지 않으면서 줄이 바뀌는 가로배치




```css
.parent {
    border: 5px solid green;
    display: flex;
    flex-wrap: wrap;
}
```

```css
.parent div {
    width: 100px;  height: 100px;
    color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;
    background-color: crimson;
}
```

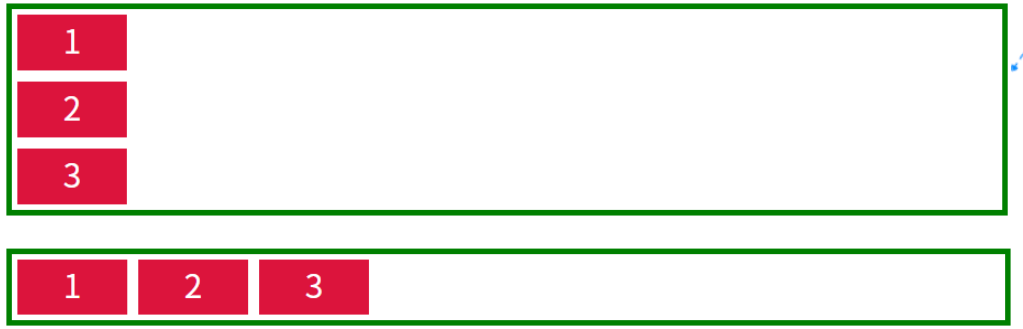
HTML

```html
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
    <div class="child3">4</div>
    <div class="child3">5</div>
    <div class="child3">6</div>
</div>
```

코딩웍스 CSS3 flexbox - 부모요소 속성

04) 부모요소 속성 중 방향과 줄 바꿈 축약형 속성 flex-flow

flex-direction 속성과 flex-wrap 속성을 flex-flow라는 축약 속성으로 합칠 수 있습니다.




```
.parent { border: 5px solid green;  display: flex;  flex-flow: column nowrap; }
.parent div {
    width: 100px;  height: 50px;  margin: 5px;  color: #fff;  text-align: center;
    font-size: 30px;  background-color: crimson;
}
```

```
.parent { border: 5px solid green;  display: flex;  flex-flow: row wrap; }
.parent div {
    width: 100px;  height: 50px;  margin: 5px;  color: #fff;  text-align: center;
    font-size: 30px;  background-color: crimson;
}
```

Row와 wrap의 순서는 관계없음

HTML

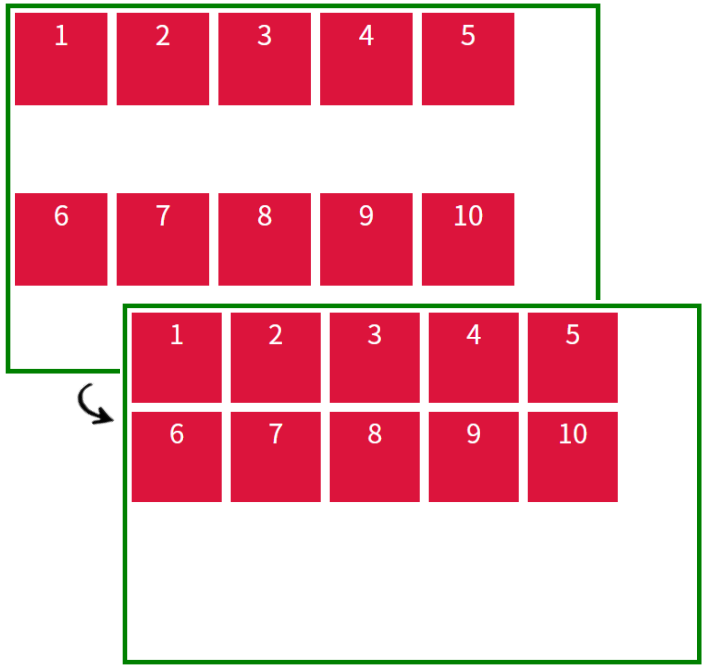
```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
</div>
```

코딩웍스 CSS3 flexbox - 부모요소 속성

05) 부모요소 사용하는 Flexbox 속성 : align-content: flex-start;

부모요소에 wrap 속성이 있는 경우 여러 개의 자식요소들의 간격 조절, 수직 수평 정렬을 변경

▼ align-content: stretch; /* 기본 값 */




```
.parent {
    border: 5px solid green;
    height: 400px;
    display: flex;
    flex-wrap: wrap;
    align-content: flex-start; /* 기본 값 */
}
.parent div {
    color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;
    background-color: crimson;
    height: 100px;  width: 100px;
}
```

flex-wrap: wrap; → 자식요소들이 줄이 바꾸는 속성

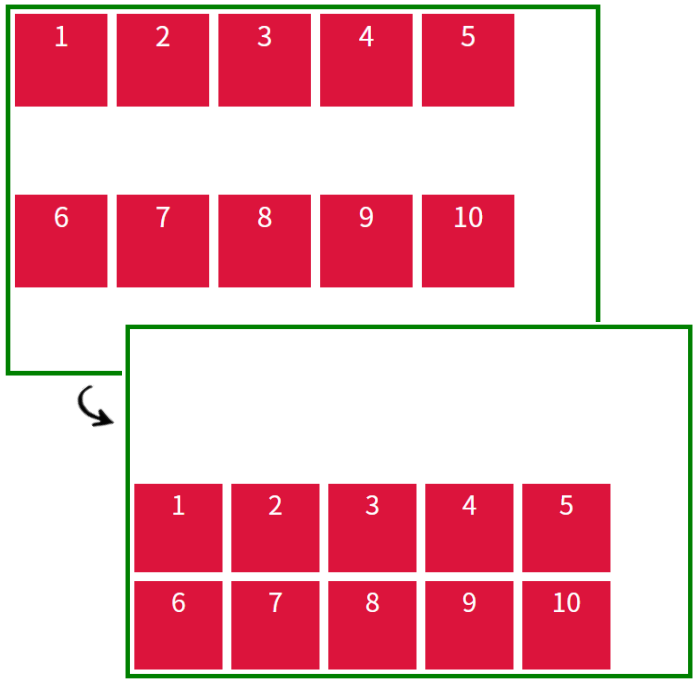
align-content: flex-start; /* 기본 값 */ → 자식요소들을 부모요소의 상단에 간격없이 뭉치기

HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
    <div class="child4">4</div>
    <div class="child5">5</div>
    <div class="child6">6</div>
    <div class="child7">7</div>
    <div class="child8">8</div>
    <div class="child9">9</div>
    <div class="child10">10</div>
</div>
```

코딩웍스 CSS3 flexbox - 부모요소 속성

05) 부모요소 사용하는 Flexbox 속성 : align-content: flex-end;




```
.parent {
    border: 5px solid green;
    height: 400px;
    display: flex;
    flex-wrap: wrap;
    align-content: flex-end;  /* 기본 값 */
}
.parent div {
    color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;
    background-color: crimson;
    height: 100px;  width: 100px;
}
```

자식요소들이 줄이 바꾸는 속성

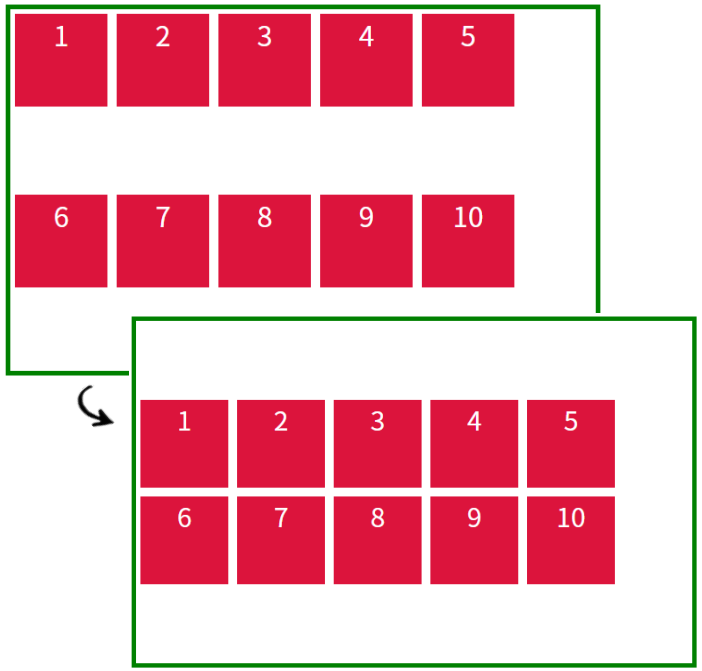
자식요소들을 부모요소의 하단에 간격없이 뭉치기

HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
    <div class="child4">4</div>
    <div class="child5">5</div>
    <div class="child6">6</div>
    <div class="child7">7</div>
    <div class="child8">8</div>
    <div class="child9">9</div>
    <div class="child10">10</div>
</div>
```

코딩웍스 CSS3 flexbox - 부모요소 속성

05) 부모요소 사용하는 Flexbox 속성 : align-content: center;




```
.parent {
    border: 5px solid green;
    height: 400px;
    display: flex;
    flex-wrap: wrap;
    align-content: center;  /* 기본 값 */
}
.parent div {
    color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;
    background-color: crimson;
    height: 100px;  width: 100px;
}
```

자식요소들이 줄이 바꾸는 속성

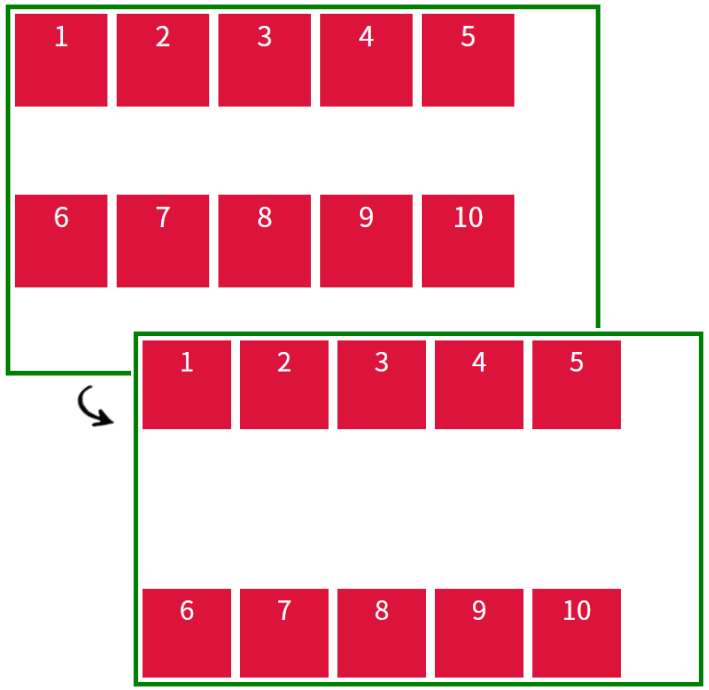
자식요소들을 부모요소의 중앙에 간격없이 뭉치기

HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
    <div class="child4">4</div>
    <div class="child5">5</div>
    <div class="child6">6</div>
    <div class="child7">7</div>
    <div class="child8">8</div>
    <div class="child9">9</div>
    <div class="child10">10</div>
</div>
```

코딩웍스 CSS3 flexbox - 부모요소 속성

05) 부모요소 사용하는 Flexbox 속성 : align-content: space-between;




```
.parent {
    border: 5px solid green;
    height: 400px;
    display: flex;
    flex-wrap: wrap;
    align-content: space-between;
}
.parent div {
    color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;
    background-color: crimson;
    height: 100px;  width: 100px;
}
```

자식요소들이 줄이 바꾸는 속성

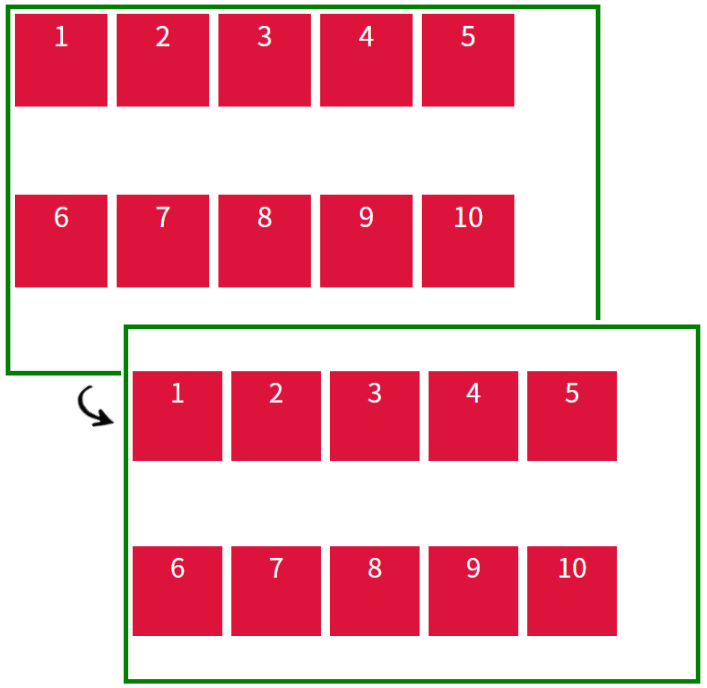
자식요소들을 부모요소의 상단과 하단으로 붙여서 뭉치기

HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
    <div class="child4">4</div>
    <div class="child5">5</div>
    <div class="child6">6</div>
    <div class="child7">7</div>
    <div class="child8">8</div>
    <div class="child9">9</div>
    <div class="child10">10</div>
</div>
```

코딩웍스 CSS3 flexbox - 부모요소 속성

05) 부모요소 사용하는 Flexbox 속성 : align-content: space-around;




```
.parent {
    border: 5px solid green;
    height: 400px;
    display: flex;
    flex-wrap: wrap;
    align-content: space-around;
}
.parent div {
    color: #fff;  text-align: center;
    font-size: 30px;  margin: 5px;
    background-color: crimson;
    height: 100px;  width: 100px;
}
```

자식요소들이 줄이 바꾸는 속성

자식요소끼리 상하 간격을 일정하게 배분해서 수평 중앙에 뭉치기

HTML

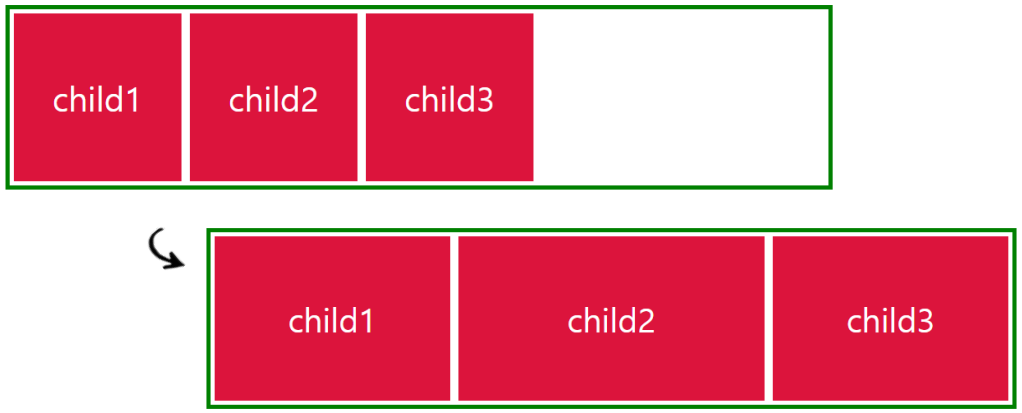
```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
    <div class="child4">4</div>
    <div class="child5">5</div>
    <div class="child6">6</div>
    <div class="child7">7</div>
    <div class="child8">8</div>
    <div class="child9">9</div>
    <div class="child10">10</div>
</div>
```

코딩웍스 CSS3 flexbox - 자식요소 속성

01-0) 자식요소 사용하는 속성 : flex-grow: 숫자(정수)

예를 들어, 요소가 3개 flex-grow를 1, 2, 1을 주었다면 비율에 따라 25%(1/4), 50%(2/4)를, 25%(1/4)을 배정 됩니다. 기본값 flex-grow: 0;

플렉스 자식요소의 증가 너비 비율을 설정



HTML

```
<div class="parent">
    <div class="child1">child1</div>
    <div class="child2">child2</div>
    <div class="child3">child3</div>
</div>
```

```
.parent {
    border: 5px solid green;
    display: flex;
}

.parent div {
    height: 200px;
    width: 200px;
    color: #fff;
    text-align: center;
    line-height: 200px;
    background-color: crimson;
}
```

```
.child1 {
    flex-grow: 1;
}
```
→ 전체의 1/4을 차지함

```
.child2 {
    flex-grow: 2;
}
```
→ 전체의 2/4을 차지함

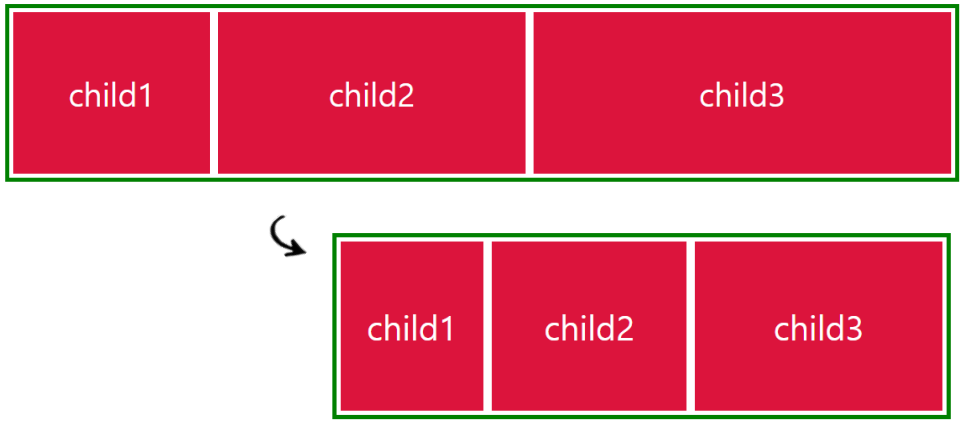
```
.child3 {
    flex-grow: 1;
}
```
→ 전체의 1/4을 차지함

코딩웍스 CSS3 flexbox - 자식요소 속성

01-0) 자식요소 사용하는 속성 : flex-shrink: 숫자(정수)

flex-shrink 요소의 너비에 영향을 받기 때문에 계산하기 복잡하고, 실전 제작에서 활용도가 많이 떨어지니 참고 정도만 하세요. 기본값 flex-shrink: 1;

플렉스 자식요소의 감소 너비 비율을 설정



HTML

```
<div class="parent">
    <div class="child1">child1</div>
    <div class="child2">child2</div>
    <div class="child3">child3</div>
</div>
```

```
.parent {
    border: 5px solid green;
    display: flex;
}

.parent div {
    height: 200px;
    color: #fff;
    text-align: center;
    line-height: 200px;
    background-color: crimson;
}
```

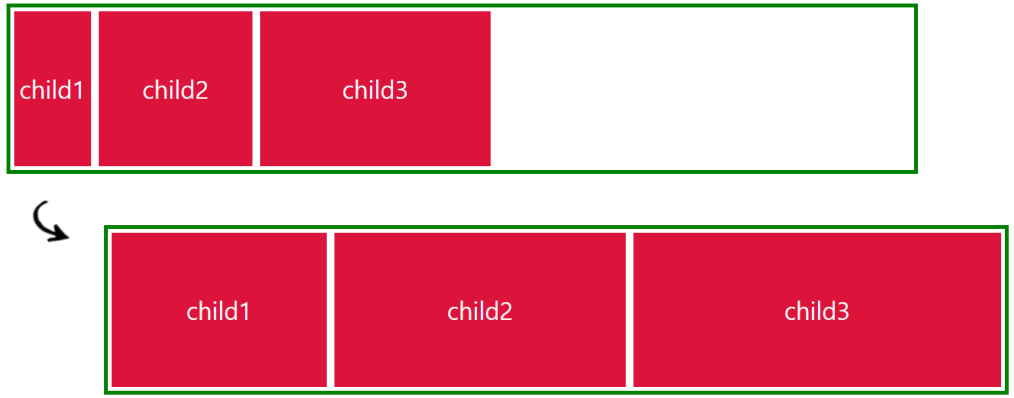
```
.child1 {
    background-color: crimson;
    flex-grow: 1;  flex-shrink: 3;
}
.child2 {
    background-color: dodgerblue;
    flex-grow: 2;  flex-shrink: 2;
}
 .child3 {
    background-color: yellowgreen;
    flex-grow: 3;  flex-shrink: 1;
}
```

코딩웍스 CSS3 flexbox - 자식요소 속성

01-0) 자식요소 사용하는 속성 : flex-basis: 숫자(정수)

flex-basis는 요소의 width, height 등의 속성으로 Item의 너비를 설정할 수 있습니다. 기본값 flex-basis: auto;

플렉스 자식요소의 공간 배분 전 기본 너비 설정



HTML

```
<div class="parent">
    <div class="child1">child1</div>
    <div class="child2">child2</div>
    <div class="child3">child3</div>
</div>
```

```
.parent {
    border: 5px solid green;
    display: flex;
}

.parent div {
    height: 200px;
    color: #fff;
    text-align: center;
    line-height: 200px;
    background-color: crimson;
    flex-grow: 1;
}
```

```
.child1 {
    background-color: crimson;
    flex-basis: 100px;
}
.child2 {
    background-color: dodgerblue;
    flex-basis: 200px;
}
 .child3 {
    background-color: yellowgreen;
    flex-basis: 300px;
}
```

코딩웍스 CSS3 flexbox - 자식요소 속성

01-1) 자식요소 사용하는 단축 속성 :

flex: [증가 너비] [감소 너비] [기본 너비]

flex: [flex-grow] [flex-shrink] [flex-basis]

```
.child {
    flex: 1 1 50px; /* 증가너비 감소너비 기본너비 */
    flex: 1 1; /* 증가너비 감소너비 */
    flex: 1 50px; /* 증가너비 기본너비 */
}
```

참고하세요~!

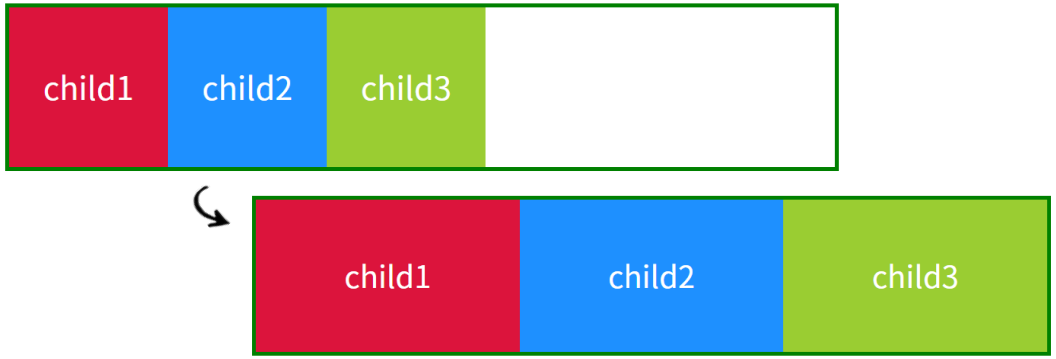
- flex 단축 속성에서 flex-grow를 제외하고 모두 생략 가능합니다.
- flex: 1; 로 작성하면 flex-grow: 1; 과 같습니다.
- flex: 1; 과 flex: 1 1; 과 flex: 1 1 0; 은 같은 의미입니다.

코딩웍스 CSS3 flexbox - 자식요소 속성

01-1) 자식요소 사용하는 단축 속성 : flex: 숫자(정수)

flex: [flex-grow] [flex-shrink] [flex-basis]
flex: 0 1 auto

플렉스 자식요소의 너비를 상대적으로 설정



HTML

```
<div class="parent">
    <div class="child1">child1</div>
    <div class="child2">child2</div>
    <div class="child3">child3</div>
</div>
```

```
.parent {
    border: 5px solid green;
    display: flex;
}

.parent div {
    height: 200px;
    width: 200px;
    color: #fff;
    text-align: center;
    line-height: 200px;
    font-size: 40px;
}
```

```
.child1 {
    background-color: crimson;
    flex: 1;
}
.child2 {
    background-color: dodgerblue;
    flex: 1;
}
.child3 {
    background-color: yellowgreen;
    flex: 1;
}
```

flex: 1; → 전체의 1/3을 차지함

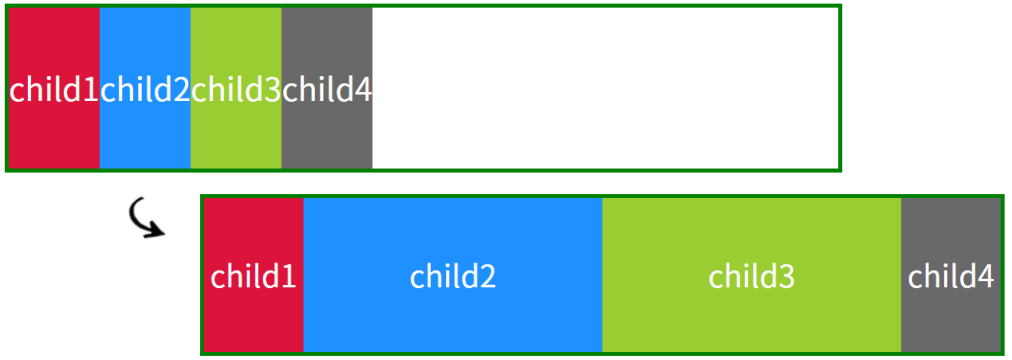
코딩웍스 CSS3 flexbox - 자식요소 속성

01-2) 자식요소 사용하는 단축 속성 : flex: 숫자(정수)

flex: [flex-grow] [flex-shrink] [flex-basis]
flex: 0 1 auto

.child1 { background-color: crimson; flex: 1; } → 전체의 1/6을 차지함
.child2 { background-color: dodgerblue; flex: 4; } → 전체의 4/6을 차지함
.child3 { background-color: yellowgreen; flex: 1; } → 전체의 1/6을 차지함

플렉스 박스 전체의 크기 계산은 1+4+1 = 6, 1+3+3+1 = 8 이 됩니다.



.child1 { background-color: crimson; flex: 1; } → 전체의 1/8을 차지함
.child2 { background-color: dodgerblue; flex: 3; } → 전체의 3/8을 차지함
.child3 { background-color: yellowgreen; flex: 3; } → 전체의 3/8을 차지함
.child4 { background-color: dimgray; flex: 1; } → 전체의 1/8을 차지함

코딩웍스 CSS3 flexbox - 자식요소 속성

01-3) 자식요소 사용하는 Flexbox 속성 : flex: 숫자(정수)

HTML

```
<div class="parent">
    <div class="child1">child1</div>
    <div class="child2">child2</div>
    <div class="child3">child3</div>
</div>
```

```
.parent {
    border: 5px solid green;
    display: flex;
}

.parent div {
    height: 200px;
    width: 200px;
    color: #fff;
    text-align: center;
    line-height: 200px;
    font-size: 40px;
}
```

```
.child1 {
    background-color: crimson;
    /* flex: 1; */
}
```

flex 값이 없으므로 width: 200px; 적용되어 고정됨

```
.child2 {
    background-color: dodgerblue;
    flex: 1;
}
```

.child1의 200px을 제외한 전체 너비의 1/2을 차지함

```
.child3 {
    background-color: yellowgreen;
    flex: 1;
}
```

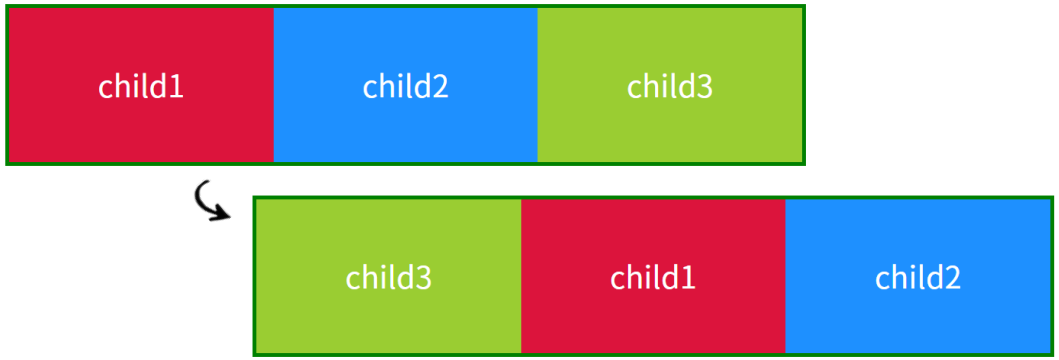
.child1의 200px을 제외한 전체 너비의 1/2을 차지함

코딩웍스 CSS3 flexbox - 자식요소 속성

02) 자식요소 사용하는 Flexbox 속성 : order: 숫자(정수)

플렉스 자식요소의 순서를 강제로 설정하는 속성인 order의 속성에 사용하는 숫자는 0, 1, 2, 3… 기본값이 0이기 때문에 order를 주지 않으면 첫번째에 배치됨.

플렉스 자식요소의 순서를 설정



HTML

```
<div class="parent">
    <div class="child1">child1</div>
    <div class="child2">child2</div>
    <div class="child3">child3</div>
</div>
```

```
.parent {
    border: 5px solid green;
    display: flex;
}

.parent div {
    height: 200px;
    width: 200px;
    color: #fff;
    text-align: center;
    line-height: 200px;
    font-size: 40px;
}
```

```
.child1 {
    background-color: crimson;
    flex: 1; order: 1;
}
```

→ 2번째 순서로 배치

```
.child2 {
    background-color: dodgerblue;
    flex: 1; order: 2;
}
```

→ 3번째 순서로 배치

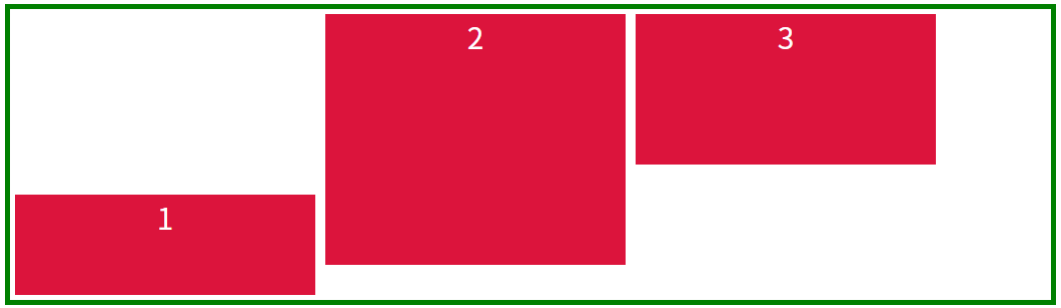
```
.child3 {
    background-color: yellowgreen;
    flex: 1;
}
```

→ order 속성을 주지 않으면 기본값이 0이기 때문에 1번째 순서로 배치됨

코딩웍스 CSS3 flexbox - 자식요소 속성

03) 자식요소 사용하는 Flexbox 속성 : align-self: flex-start | flex-end | center | baseline | stretch

부모요소의 align-items와 별개로 겨별 자식요소들의 상단, 중앙, 하단 등 수직배치 변경



HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
</div>
```

```
.parent {
    border: 5px solid green;  display: flex;
    height: 500px; align-items: flex-start;
}
.parent div {
    width: 300px;  color: #fff;  text-align: center;   margin: 5px;
    background-color: crimson;
}
.child1 { height: 100px;  align-self: flex-end; }
.child2 { height: 350px; }
.child3 { height: 250px; }
```

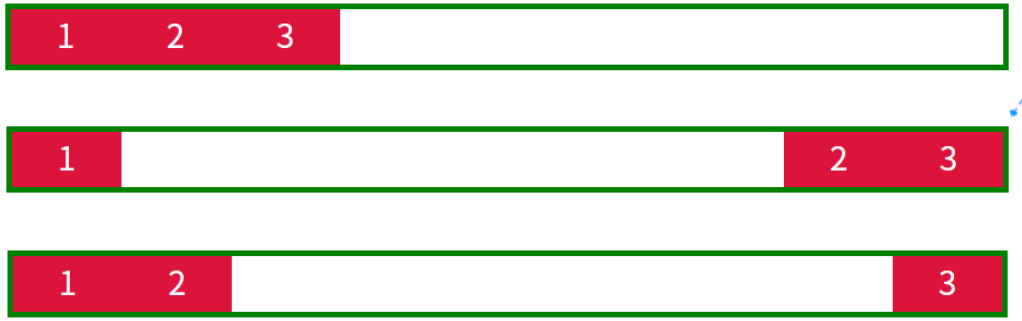
부모요소에 수직 정렬 align-items를 적용

자식요소에 수직 정렬을 align-sefl로 개별적으로 적용

코딩웍스 CSS3 flexbox - 자식요소 속성

03) Flexbox 자식요소 사용하는 마진 속성(1) : margin: auto; margin-right: auto; margin-right: auto;

margin 속성을 auto로 설정해서 타 요소를 반대방향으로 밀어내 왼쪽 끝 또는 오른쪽 끝에 배치




```
.parent { border: 5px solid green;  display: flex; }
.parent div { width: 100px;  height: 50px;  color: #fff;  text-align: center;
    font-size: 30px;  background-color: crimson;
}
.child1 { margin-right: auto; }
```

```
.parent { border: 5px solid green;  display: flex; }
.parent div { width: 100px;  height: 50px;  color: #fff;  text-align: center;
    font-size: 30px;  background-color: crimson;
}
.child3 { margin-left: auto; }
```

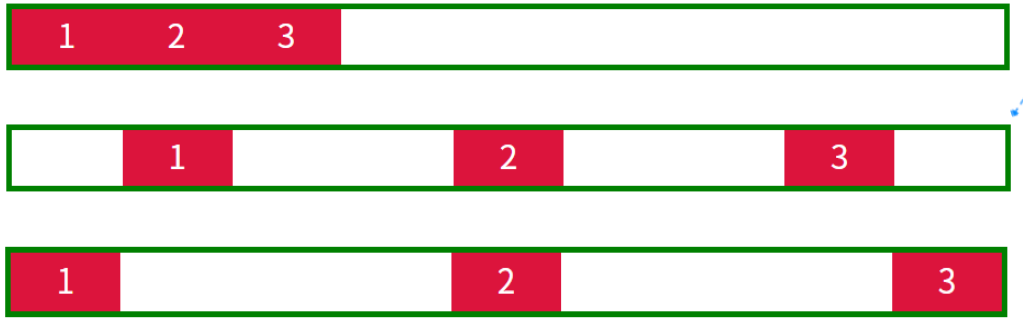
HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
</div>
```

코딩웍스 CSS3 flexbox - 자식요소 속성

03) Flexbox 자식요소 사용하는 마진 속성(2) : margin: auto; margin-right: auto; margin-right: auto;

margin 속성을 auto로 설정해서 타 요소를 반대방향으로 밀어내 왼쪽 끝 또는 오른쪽 끝에 배치




```
.parent { border: 5px solid green;  display: flex; }
.parent div { width: 100px;  height: 50px;  color: #fff;  text-align: center;
    font-size: 30px;  background-color: crimson;
}
.child1, .child2, .child3 { margin: auto; }
```

```
.parent { border: 5px solid green;  display: flex; }
.parent div { width: 100px;  height: 50px;  color: #fff;  text-align: center;
    font-size: 30px;  background-color: crimson;
}
.child2 { margin: auto; }
```

HTML

```
<div class="parent">
    <div class="child1">1</div>
    <div class="child2">2</div>
    <div class="child3">3</div>
</div>
```

코딩웍스 CSS3 flexbox - 플렉스 응용 예제

응용 예제 #01) 수직중앙 수평중앙 중첩해서 사용하기

부모요소가 자식요소를 수직중앙 수평중앙에 위치시킬 때 사용하는 플렉스 구문

HTML

```
<div class="container">
    <section>
        <h1>수직중앙 수평중앙 중첩하기</h1>
    </section>
</div>
```

```
body {
    height: 100vh;
    display: flex; justify-content: center; align-items: center;
}
.container {
    display: flex; justify-content: center; align-items: center;
}
section {
    display: flex; justify-content: center; align-items: center;
}
article h1 { font-size: 2em; }
```

display: flex; → 플렉스 시작설정
justify-content: center; → 자식요소를 부모요소의 수평중앙
align-items: center; → 자식요소를 부모요소의 수직중앙

```
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
```

포지션 속성으로 센터링(Centering) 해도 동일한 결과

```
display: flex;
justify-content: center;
align-items: center;
```

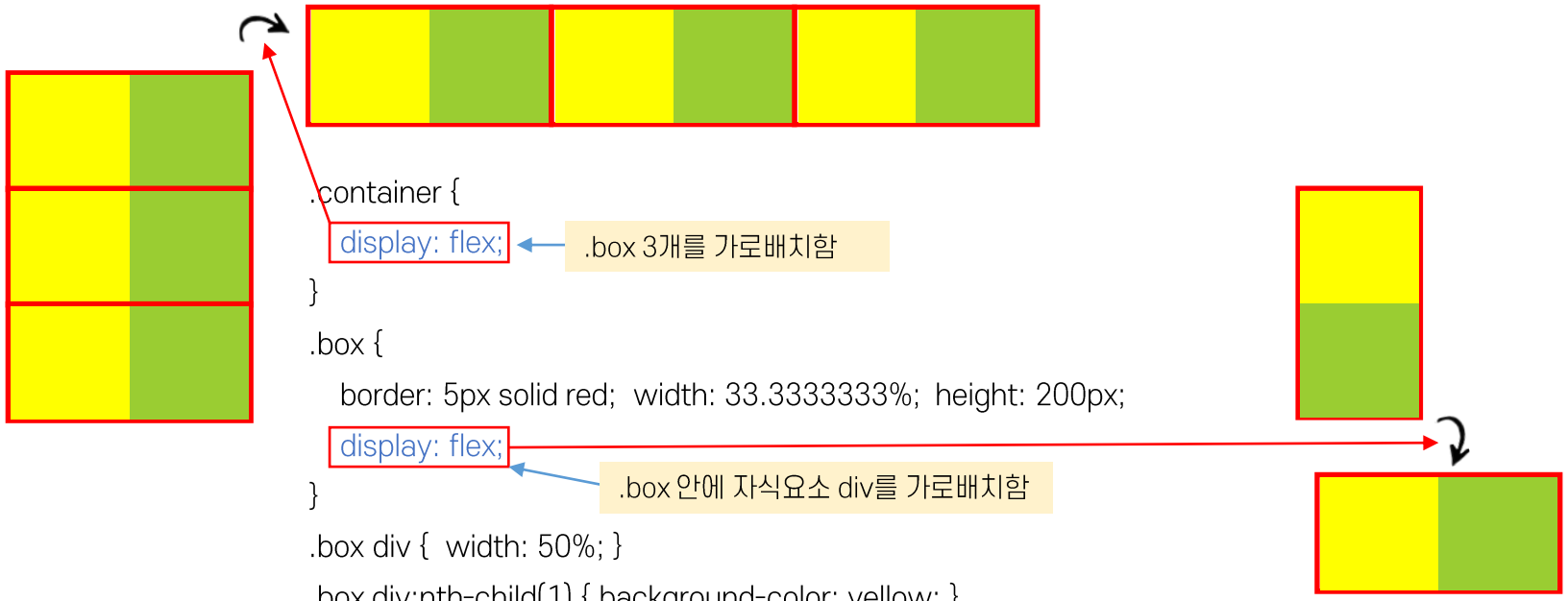
자식요소를 부모요소의 수직중앙 수평중앙에 오게끔 만드는 플렉스 구문 세트 (계속 중첩해서 사용할 수 있음)

코딩웍스 CSS3 flexbox - 플렉스 응용 예제

응용 예제 #02) 부모자식으로 중첩된 div를 가로 배치하기

HTML

```
<div class="container">
    <div class="box">
        <div></div>
        <div></div>
    </div>
    <div class="box">
        <div></div>
        <div></div>
    </div>
    <div class="box">
        <div></div>
        <div></div>
    </div>
</div>
```

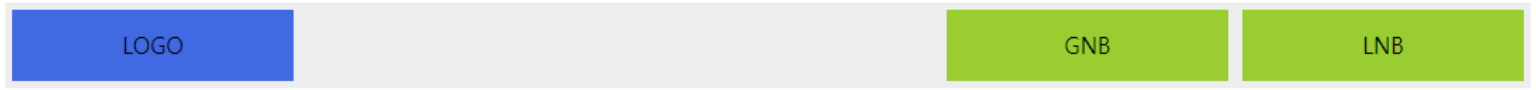


코딩웍스 CSS3 flexbox - 플렉스 응용 예제

응용 예제 #03) 상단 네비게이션 만들기

HTML

```
<header>
    <div class="logo">logo</div>
    <div class="gnb">gnb</div>
    <div class="lnb">lnb</div>
</header>
```

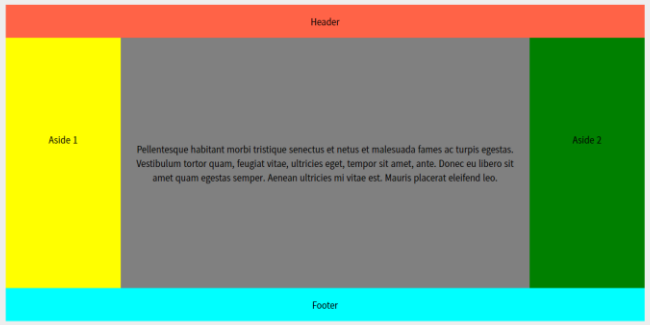



```
header {
    background-color: #eee; display: flex;
}
header div {
    width: 200px;  height: 50px;  background-color: yellowgreen;
    margin: 5px;  text-align: center;  line-height: 50px;
    text-transform: uppercase;
}
.logo {
    background-color: royalblue;
    margin-right: auto;
}
```

header 밑에 div 3개를 가로배치함

.logo의 오른쪽 마진을 auto로 주어서 나머지 div를 오른쪽으로 밀착시킴

코딩웍스 CSS3 flexbox 플렉스 응용 예제 #04) 시멘틱 태그 반응형 레이아웃



→ body

HTML

```
<div class="container">
  <header>Header</header>
  <section>
    <article class="main">
      <p>...</p>
    </article>
    <article class="left">Aside 1</article>
    <article class="right">Aside 2</article>
  </section>
  <footer>Footer</footer>
</div>
```

```
.container {                                   ①
  width: 1200px; margin: auto;  text-align: center;
  display: flex;
  flex-wrap: wrap;  → 너비가 컨텐츠 보다 작아 졌을 때 줄바꿈
}
header {
  background-color: tomato;  padding: 1em;
  flex: 1;  → 100%로 전체 너비
}
section { display: flex; }  → 자식요소 article 가로배치
section article {  padding: 15em 1em;  }
.main {
  background-color: gray; flex: 4; order: 2;
}                          4/6 너비를 가짐   2번째 순서
.left {
  background-color: yellow; flex: 1; order: 1;
}                           1/6 너비를 가짐   1번째 순서
```

```
.right {
  background-color: green;
  flex: 1; order: 3;  → 3번째 순서
}                 1/6 너비를 가짐
footer {
  background-color: aqua;
  padding: 1em;
  flex: 1;  → 100%로 전체 너비
}
@media (max-width: 768px) {                    ②
  .container { width: 100%; }
  section {
    flex-direction: column;
  }  모바일에서 article 3개 세로로 배치
  section article {  padding: 1em; }
  .main { order: 1; }
}                  모바일에서 1번째 순서로 변경
```

.container

① PC에서는 1200px 로 고정되어 있고,
② 모바일에서 100% 로 변경

모바일 화면에서 실행 할 미디어쿼리 CSS