

CSS calc() / min() / max() / clamp() Test Page

1 — Basic Arithmetic



addition
width: calc(100px + 40px);



subtraction
width: calc(200px - 60px);



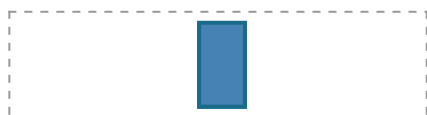
multiplication
width: calc(20px * 4);



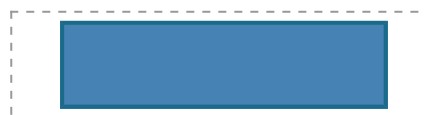
division
width: calc(200px / 4);

2 — Mixed Units & Percentages

calc(1em + 5px) resolves the em against the element's own font-size; calc(100% - 40px) resolves the percentage against the 200px dashed container below.



1em + 5px @ 16px font
width: calc(1em + 5px)



100% - 40px in a 200px container
width: calc(100% - 40px)

3 — Nested calc() and Parentheses



nested calc()
width: calc(calc(50px + 50px) * 2);



parenthesized grouping
width: calc((50px + 50px) * 2);

4 — margin / padding / border-radius / height



margin-left
margin-left: calc(20px + 10px); width: 60px;



padding (widens box)
padding: calc(5px + 5px); width: 60px;



border-radius
border-radius: calc(10px + 10px); width: 80px;



height
height: calc(20px + 20px); width: 80px;

5 — Negative Result

PeachPDF doesn't clamp a negative calc() result to zero, matching how a plain negative length is already handled.

calc(50px - 100px)
width: calc(50px - 100px); height: 20px; border: 1px dashed red;

6 — min() / max() / clamp()



min(150px, 100px)
width: min(150px, 100px);



max(150px, 100px)
width: max(150px, 100px);

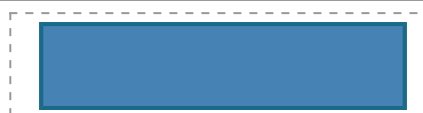


clamp(50px, 300px, 150px)
width: clamp(50px, 300px, 150px);



clamp(50px, 10px, 150px)
width: clamp(50px, 10px, 150px);

7 — calc() Combined With a Custom Property



calc() referencing a custom property
width: calc(100% - var(--gap))