

GSAP cheat sheet

Every bit of syntax from the GSAP crash course in the order we learned it.

gsap@3.13.0

Free with all the extra tools

■ Setup

Chapter 2

Scripts go at the bottom of the body so the page loads first. GSAP first and your own code right after it.

```
<script src="https://cdn.jsdelivr.net/npm/gsap@3.13.0/dist/gsap.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/gsap@3.13.0/dist/ScrollTrigger.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/gsap@3.13.0/dist/SplitText.min.js"></script>
<script>
  gsap.registerPlugin(ScrollTrigger);
  gsap.registerPlugin(SplitText);
  console.log(gsap.version);
</script>
```

gsap.min.js	GSAP itself so it always loads first	gsap.registerPlugin()	tells GSAP the plugin exists so it can use it
ScrollTrigger.min.js	a plugin and only the file name changes	console.log(gsap.version)	prints 3.13.0 when GSAP is loaded

Watch out gsap is not defined means your script sits above the GSAP one. Move it below and refresh.

■ Your first tween

Chapter 3

A tween is one move from A to B. This one moves every box 300 pixels right over one second.

```
gsap.to(".box", { x: 300, duration: 1 });
```

gsap.to	animate the target to the values we give it
".box"	the target and it grabs every element with the class box
{ }	the settings object with a comma between each setting
x: 300	300 pixels to the right and a minus number goes left
duration: 1	one second and the default is 0.5

Watch out duration: 1000 is almost 17 minutes because GSAP counts in seconds. Write 1 for one second.

■ Properties

Chapter 4

The same gsap.to line every time. Only the settings inside the braces change.

```
{ x: 300, y: 200, rotation: 360, scale: 1.5, opacity: 0 }
{ backgroundColor: "#ff8a00", xPercent: 100 }
```

x y	move in pixels and a bigger y moves down
rotation	spins in degrees so 360 is one full turn
scale	1 is the normal size so 2 is twice as big
opacity	0 is invisible and 1 is fully visible
backgroundColor	CSS names drop the dash and get a capital letter
xPercent yPercent	move by a percent of the box's own size

Watch out left: 300 needs a position and can look jumpy. Use x and y to move things and keep left and top for layout.

■ to, from, fromTo and set

Chapter 5

Four ways to say where an animation starts and where it ends.

```
gsap.to(".box", { x: 300, duration: 1 });
gsap.from(".box", { y: 100, opacity: 0 });
gsap.fromTo(".box", { x: 0 }, { x: 300 });
gsap.set(".box", { x: 100 });
```

to	go to here from wherever it is now
from	start at these values and end where it sits
fromTo	you pick the start and the end and duration goes in the end
set	jumps there straight away with no animation

Watch out A from flashes on load? Hide it in CSS with visibility: hidden and set it to "visible" right before the from.

■ Timing

Chapter 6

When it starts and how many times it plays. Delays are in seconds.

```
{ delay: 0.5, repeat: -1, yoyo: true }
{ repeat: 2, repeatDelay: 0.5 }
```

delay	how long GSAP waits before it starts
repeat: 2	two extra rounds so it plays three times
repeat: -1	never stop which is great for loaders
yoyo: true	every other round plays backwards
repeatDelay	a short rest between every round

Watch out Repeat on its own snaps back to the start. Add yoyo: true and it glides back.

■ Eases

Chapter 7

An ease is how a move speeds up and slows down. Time goes along the bottom and distance goes up the side.



```
ease: "power2.out"
ease: "back.out"
```

.out	slows down at the end of the move
.in	starts slow and speeds up
.inOut	does both

power1 power4 the number is how strong it is
"power2.out" a string so it goes in quotes

Watch out Elastic on everything makes the whole page wobble. Use power2.out for most things and save back or elastic for one small fun moment.

■ Stagger

Chapter 8

When the target matches lots of things each one starts a little after the one before.

```
gsap.from(".card", { y: 100, opacity: 0, stagger: 0.1 });
stagger: { each: 0.05, from: "center", grid: "auto" }
```

stagger: 0.1	card two starts 0.1 seconds after card one
each	the gap between every item
from	where the wave begins like "center" or "edges" or "end" or "random"
grid: "auto"	they sit in rows and columns so GSAP finds the real middle
"centre"	the British spelling is ignored so write center

Watch out No forEach loop with a delay for every card. One gsap.from with stagger does the same thing.

■ Timelines

Chapter 9

A timeline plays tweens in order like a playlist. The position parameter goes third and says when a step starts.

```
const tl = gsap.timeline();
tl.from(".logo", { scale: 0 })
  .from(".title", { y: 40, opacity: 0 }, "-=0.3")
  .addLabel("text")
  .from(".btn", { y: 20, opacity: 0 }, "text+=0.2");
```

"-=0.3"		starts 0.3 seconds early so it overlaps
"<"		starts together with the one before
"+=0.5"		leaves a half second gap
"text"		starts right at the label called text
"text+=0.2"		0.2 seconds after that label

Watch out A delay on every tween means fixing every delay by hand. In a timeline everything moves along by itself.

■ Controls and events

Chapter 10

paused: true makes it wait and the methods let you drive it from any event.

```
const tl = gsap.timeline({ paused: true });
tl.play(); tl.pause(); tl.reverse(); tl.restart();
const play = document.querySelector(".play");
play.addEventListener("click", () => tl.play());
```

reverse()	plays it back from where it is
restart()	jumps to the start and plays again
onComplete	runs a function when the timeline finishes
querySelector	finds the first match on the page
addEventListener	listens for "click" or "mouseenter" or "mouseleave"
() =>	an arrow function that runs on every event

Watch out A new tween on every hover gets stuck half faded. Build the timeline once and just play and reverse it.

■ ScrollTrigger

Chapter 11

Watches your scroll and tells an animation when to play. Small s is the setting and big S is the plugin.

```
gsap.to(".box", {
  x: 300,
  scrollTrigger: { trigger: ".box", start: "top center" }
});
```

trigger	the element we watch
start: "top center"	a spot on the box then a spot on the screen
end: "bottom center"	works the same way
markers: true	draws the lines so green is start and red is end
toggleActions	four words for four moments and "play none none reverse" rewinds when you scroll back up

Watch out Plays right away and ignores your scroll? The plugin never got loaded or registered.

■ Scrub and pin

Chapter 11

With scrub the scroll between start and end becomes the duration so we drop duration.

```
scrollTrigger: {
  trigger: ".section", start: "top top", end: "+=1000",
  scrub: true, pin: true
}
```

scrub: true	glues the animation to your scrollbar
scrub: 1	takes one second to catch up so it feels smoother
ease: "none"	keeps it at the same speed as your scroll
pin: true	keeps a section stuck to the screen while you scroll
end: "+=1000"	1000 pixels of scrolling after the start

Watch out Markers are just for building. Delete that line before you share your page.

■ SplitText

Chapter 12

Cuts text into lines or words or letters so each one can move.

```
const split = SplitText.create(".title", {
  type: "lines, words",
  mask: "lines"
});
gsap.from(split.lines, { yPercent: 100, stagger: 0.1 });
```

type	what to cut it into and add chars for every letter
split.lines	a list of every line and split.words gives every word
mask: "lines"	puts each line in a little window that hides what sticks out
yPercent: 100	starts one full line down and out of sight

Watch out Lines break in weird places when the font is still loading. Wrap it all in document.fonts.ready.then(() => {}).

■ Bonus: the image loader

Chapter 13

Everything together in one timeline. The counter hits 100 right as the last photo takes over.

```
window.addEventListener("load", () => {
  document.fonts.ready.then(() => {
    const tl = gsap.timeline();
  });
});
```

clip-path: inset(50% 0% 50% 0%)	closes each photo down to a line across the middle
clipPath: "inset(0% 0% 0% 0%)"	opens it all the way round
stagger: { amount: 2 }	spreads all six starts across 2 seconds
yPercent: -2000	rolls a column of digits up 20 numbers
.addLabel("reveal")	pins a label at 100 so the rest is timed from there