

A LIFE IN HABITS

Jensen Huang

The Discipline Behind the Man Who Built NVIDIA

A short study of one man's routines, mindset, and way of working — not his net worth. What he does every day, how he thinks, how he leads, and what any of us can actually borrow from it.

Why Study This Man At All

01

Jensen Huang co-founded NVIDIA in 1993 and, three decades later, turned it into one of the most valuable companies on Earth. That fact alone tells you almost nothing useful. Plenty of founders got rich and stayed shallow. What makes Huang worth studying is not the size of the company — it's the size of the discipline underneath it. The same habits that carried him through thirty years of near-failure are still running the show today, largely unchanged.

This isn't a book about valuations or stock charts. You won't find advice here about getting rich. What you will find is a close look at how one person structured his days, treated his failures, ran his meetings, and kept working long after most people would have quit or coasted. Huang has said the feeling of nearly going out of business "doesn't leave you" — and that feeling, more than any single decision, is what shaped how he operates.

The pages ahead move in two halves. First, the life: a childhood spent being sent away, a diner job, a company that came within weeks of collapsing more than once. Second, the habits: the daily routine, the leadership rules, the way he handles mistakes and mentors people. Read it as a study of a working method, built by someone who has had to defend it under real pressure for a very long time.

It also helps to say plainly what this account is not. It isn't a growth-hacking guide, and it isn't a lecture on chasing valuation. Where most write-ups about Huang default to net worth and stock price, this one deliberately skips that math. The interesting number here was never the market cap — it was the number of mornings he kept showing up before anyone was watching.

Fast fact: NVIDIA started in 1993 with about \$40,000 in seed money. Three decades later, it became one of the most valuable companies on the planet.

The Boy Who Was Sent Away Twice

02

Jensen Huang was born in Tainan, Taiwan. When he was five, his family relocated to Thailand. Four years later, with political unrest building, his parents made a decision that would shape everything that followed: they sent nine-year-old Jensen and his older brother to live with an uncle in the United States, hoping to get their sons somewhere stable while the family situation in Asia stayed uncertain.

There was a mix-up. The boys landed not in a mainstream American school but at a small, strict boarding school in rural Kentucky — a place that, it turned out, was more reform school than prep school. Jensen was the only Asian student. He was assigned the top bunk above a boy covered in knife scars, tasked with cleaning toilets as a daily chore, and left to figure out an unfamiliar country largely on his own, without his parents nearby and without any real support system.

It is tempting to skip past this part of the story to get to the technology. Don't. A nine-year-old dropped into an unfamiliar country, doing manual chores in a place he did not choose, is the actual starting condition for everything that came later. The habits of self-reliance and low expectations that people now describe as "the Huang way" did not start in a boardroom. They started in a dorm room in Kentucky.

It also explains something people miss about his later leadership style. Someone who spent a year navigating an unfamiliar, occasionally hostile environment with no adult reliably in his corner tends to build an internal compass early, because external structure can't always be trusted to show up on time. That compass, arguably, is still doing a lot of the work today.

The sense of vulnerability, the sense of uncertainty, the sense of insecurity — it doesn't leave you.

JENSEN HUANG, ON CARRYING EARLY HARDSHIP FORWARD

Oregon, and a Job Bussing Tables

03

Two years after Jensen arrived in Kentucky, his parents reunited with their sons and settled in Beaverton, Oregon. He enrolled at Aloha High School, moved quickly through his coursework, and graduated early, around age sixteen. Academically, he was already ahead. Personally, he still needed a job, and the one he took was at a local Denny's, first as a busboy, later working graveyard shifts as a waiter.

It's a detail people like to mention and then rush past, but it deserves a moment. Wiping down tables and taking orders through the night shift is not glamorous work, and it doesn't teach you how to build a microchip. What it does teach, reliably, is how to show up on a schedule you don't control, how to deal with people who are tired or short-tempered, and how to keep moving when a shift feels endless. Those are unremarkable skills. They are also the exact skills that a company running on fumes needs from its founder, years later.

Huang went on to Oregon State University for a degree in electrical engineering, then to Stanford for a master's. Nothing about that path was unusual for an engineer of his generation. What was unusual was the years right before it — the ones spent clearing plates — which he has referenced often enough that it's clearly not a throwaway anecdote to him. It's a reference point.

It's also worth noting what that period didn't include: no start-up mentors, no privileged network, no early exposure to Silicon Valley. Just a work shift, a school schedule, and whatever discipline he supplied himself. Much of the infrastructure most founders lean on today simply wasn't there yet for him.

Fast fact: Huang graduated high school around age sixteen, then earned an electrical engineering degree at Oregon State before a master's at Stanford.

A Company Started in a Diner Booth

04

In 1993, Huang co-founded NVIDIA with Chris Malachowsky and Curtis Priem. The founding meetings, as the story is usually told, happened at a Denny's in San Jose — the same restaurant chain where Huang had once worked the night shift as a teenager, though a different location. They started with about forty thousand dollars and a bet that computer graphics would become important enough to deserve chips built specifically for the job.

There was no guarantee of that in 1993. Graphics chips were a crowded, unglamorous corner of the industry, and dozens of competitors were chasing the same idea. NVIDIA's first product line was not a success. The company had misjudged where the graphics standard was heading, and the resulting chip did not sell the way it needed to. This is the part of the story that gets skipped in most retellings, and it matters more than the founding date does: the company's first real product was a mistake, and everyone involved knew it.

What separates this story from a thousand failed startups is not that NVIDIA avoided that mistake. It didn't. It's what happened next, with the company's cash running out and one very uncomfortable phone call left to make.

The three founders were, by their own account, not chasing a guaranteed win. Graphics acceleration was a real but unproven bet, up against better-funded rivals and no settled standard for how 3D graphics would even be built. Betting on a narrow technical thesis, with no fallback plan, was the founding condition — not a footnote to it.

Fast fact: NVIDIA's founding meetings reportedly happened at a Denny's in San Jose — the same chain where Huang once worked as a teenage waiter in Oregon.

Thirty Days From Going Out of Business

05

By the mid-1990s, NVIDIA was building a graphics chip for Sega's next game console, under contract, with a final payment due on delivery. Partway through, Huang's team realized their approach was flawed — the chip would not work the way Sega needed. The company had roughly thirty days of cash left. The disciplined, honest move, and also the terrifying one, was to fly to Japan and tell Sega's CEO the product would not work, and that the deal should be cancelled.

Huang made that call anyway. He told Sega the truth, and separately asked whether the company could still receive the final five-million-dollar payment it was contractually owed, despite killing its own deliverable. Sega agreed, structuring it as an investment rather than a straight payment. That money became the bridge that let NVIDIA build its next chip, the RIVA 128, which shipped in 1997 and sold a million units in four months. That chip is what actually saved the company.

Huang has referred to being "thirty days from going out of business" for over thirty years since. Not because it's a good story to tell at conferences, but because, by his own account, the feeling never fully faded. It's worth sitting with the sequence here: the honest disclosure of failure came before the financial rescue, not after. That ordering is the entire lesson.

It's worth naming what didn't happen here, too. Huang didn't quietly ship the flawed chip and hope no one noticed, and he didn't wait for Sega to discover the problem on its own. The uncomfortable disclosure came first, by choice, while the company still had something real left to lose by being honest.

Suffering is part of the journey. You will appreciate it so much more when things do go well.

JENSEN HUANG

Why He Wishes You Hardship

06

In a commencement address, Huang told graduating students he wished them "ample doses of pain and suffering." Read quickly, that sounds like a strange thing to say to twenty-two-year-olds starting their careers. Read carefully, it's a specific claim about how resilience actually forms, and it's worth taking seriously rather than treating as a soundbite.

His argument, made in several interviews since, is that character and culture are not built during easy stretches. They're built when a company — or a person — is up against genuinely bad odds, with no external reason things should work out. NVIDIA's near-collapse in the 1990s is his standing example: the culture that exists at the company today, he argues, was set in that period of real danger, not in the calmer years that followed success.

This is a useful correction to a common instinct, which is to protect ourselves and the people we manage from hardship whenever possible. Huang's position is closer to the opposite: hardship, absorbed early and survived, is what calibrates a person's or a team's sense of what they can actually handle. The goal isn't to manufacture suffering. It's to stop being surprised by it, and to stop treating its arrival as evidence that something has gone wrong.

It's a view echoed, in different words, across a lot of leadership research on adversity: comfort doesn't reveal what a team or a person can actually absorb, because comfort was never testing that capacity in the first place. Only real pressure does. Huang's version simply states that plainly, without softening it into a slogan.

Takeaway: Difficulty you survive recalibrates what you believe you can handle next time. Treat hard periods as data, not just damage.

What an Ordinary Day Looks Like

07

By most accounts from people who've worked alongside him, Huang's day starts early, well before sunrise, and includes deliberate physical activity — a workout or a run — before the workday properly begins. That ordering is not incidental. Starting the body before starting the inbox is a small, repeatable act of control at the one point in the day when almost nothing else is competing for attention yet.

What follows is not a loosely structured day. Huang has described planning his priorities early and deliberately, rather than letting the first email or the loudest person set the day's agenda. He also protects time for reading — research papers, industry reports, material well outside NVIDIA's own product lines — on the reasoning that a person running a technology company has to keep learning at the same rate the technology itself is changing, or fall behind his own company.

None of this is exotic advice. Wake early, move your body, decide your priorities before the day decides them for you, keep learning deliberately. The value here isn't novelty. It's that someone running a company at enormous scale has kept a version of this simple structure for decades, under real pressure, rather than letting success talk him out of it.

None of this requires imitating his exact hours. The transferable piece is the ordering: body, then priorities, then inbox — in that sequence, not reversed. Most people default to checking messages first thing, which quietly hands the day's agenda to whoever happened to email overnight.

Fast fact: Huang sets his day's priorities before opening email — so his own agenda decides the day, instead of whoever messages him first.

Seven Days a Week — And Why That's Not Simply Advice

08

Huang has openly said he works seven days a week, including holidays, and has described the feeling behind it as a constant, low-grade anxiety about the company's survival — not calm confidence. This detail matters because it's usually flattened into generic "hustle" advice, which misses what he actually said. He isn't describing a system he'd necessarily recommend. He's describing a habit that formed under real threat and never fully switched off, even once the threat receded.

It is worth being honest about the trade-off here rather than admiring it uncritically. A schedule like this has a cost, and Huang has acknowledged that little separates his working hours from his personal hours. The useful takeaway isn't "work every day forever." It's narrower and more honest: sustained intensity is often driven by a specific fear or memory, not by willpower alone, and it's worth knowing what is actually driving your own intensity before you decide whether to keep it or change it.

The more transferable piece is the underlying discipline, not the specific hour count: treating the business as something that could still fail, long after the evidence suggests otherwise. That mental posture — staying slightly uncomfortable on purpose — is different from simply logging more hours, and it's the part actually worth borrowing.

Worth stating directly: this is the part of Huang's routine that comes with the clearest health caveat. Chronic, unmanaged stress carries real costs, and admiration for the results shouldn't blur into admiration for the method. Read this page as a caution as much as a case study.

Fast fact: Huang has said the anxiety of nearly failing has stayed with him for over thirty years — even after NVIDIA became one of the world's most valuable companies.

No One-on-Ones, on Purpose

09

Huang runs NVIDIA with around sixty direct reports and, by design, holds no routine private one-on-one meetings with any of them. For most executives, that number of direct reports alone would be considered unmanageable. Refusing private check-ins on top of that looks, on paper, like a mistake. Huang's reasoning is the opposite of careless: he calls it "reasoning in public."

The idea is that when a leader gives feedback, works through a hard decision, or corrects a mistake in a private one-on-one, only one person benefits from watching how that reasoning unfolds. Do the same thing in front of the full leadership group, and dozens of people get to see the thinking, not just the conclusion. Huang has said he'd rather have sixty people who each know what he knows than six people who each know a tenth of it.

This isn't a claim that private meetings are inherently bad. It's a specific bet about where the real bottleneck sits in a fast-moving company: not in decisions, but in how slowly good judgment normally spreads through an organization. Reasoning out loud, in front of the room, is one way to make that spread faster.

It also changes what a meeting is for. Rather than a status update delivered upward, a meeting becomes a place where the whole room watches a real decision get made, mistakes included. That's a very different use of everyone's time than the version most companies default to.

I would rather have sixty people who all know what I know than six people who each know a tenth of what I know.

JENSEN HUANG

Flat by Design, Not by Accident

10

Most large companies add layers of management as they grow, on the theory that no single person can stay informed about everything past a certain size. Huang has resisted that instinct directly. NVIDIA's structure stays deliberately flat, with information flowing to and from a wide leadership group rather than filtering down through several management tiers.

His stated reasoning is that hierarchy doesn't just slow decisions down — it quietly distorts information as it passes through each layer, with each manager summarizing, softening, or reframing what they heard before passing it along. By the time something reaches the top, or reaches the front line, it may barely resemble what was originally said. A flatter structure isn't about being informal. It's an attempt to keep information intact.

The practical result, according to people who've studied NVIDIA's operating style, is a company that can change direction faster than its size would suggest it should be able to. Huang describes this as a kind of continuous planning: watching, orienting, deciding, and acting faster than the market shifts around you, rather than waiting for an annual planning cycle to catch up. Speed, in his framing, is mostly a byproduct of clean information, not of urgency alone.

It's a structure that only works, by most accounts, because Huang is willing to sit through long, unfiltered meetings himself rather than delegating that exposure away. Flatness isn't free. Someone has to absorb the volume of raw information it produces, and here, that someone is the CEO.

Fast fact: NVIDIA runs with roughly 60 direct reports to Huang inside a company of over 30,000 people — an unusually flat structure for its size.

Owning Mistakes Loudly, on Purpose

11

When NVIDIA shipped a badly flawed graphics chip in the early 2000s — a well-documented stumble known internally and publicly as a low point — Huang did not manage the fallout quietly. He addressed the failure directly and openly rather than containing the conversation to a small group. The instinct in most companies is to protect a struggling employee, or a struggling leader, from public scrutiny. Huang's instinct runs the other way on purpose.

His reasoning: if one person's mistake gets corrected privately, only that person learns from it. If the same mistake gets discussed openly, an entire organization learns from a single failure instead of waiting to make their own version of it later. It's a deliberately uncomfortable trade — more embarrassment for the individual, in exchange for less repeated failure across the company.

This only works, worth noting, alongside the other half of his management style: he has also said he is slow to give up on people, preferring, in his words, to "torture them into greatness" rather than replace them quickly. Public accountability and personal patience aren't in tension in his approach — they're presented as two halves of the same idea. Correct the mistake in the open, but don't discard the person who made it.

This runs directly against a common instinct in management training, which usually favors private, gentle correction to preserve dignity. Huang's bet is that the dignity cost to one person is worth paying if it measurably shortens how long the same mistake survives across a 30,000-person company.

Fast fact: Huang has said he'd rather "torture people into greatness" than fire them quickly — public accountability paired with real patience.

Explaining the "Why," Every Time

12

A recurring theme in how former and current NVIDIA employees describe Huang is that he narrates his own thinking constantly — not just the decision he's made, but the full reasoning that led to it, including the parts he was unsure about. In meetings, this means explaining why an option was rejected, not only which option was chosen.

The value of this habit compounds slowly rather than immediately. A team that only hears final decisions can follow instructions, but it can't anticipate the next decision, because it never learned the underlying logic. A team that regularly hears the reasoning starts to internalize the decision-making pattern itself, and can eventually make similar calls without needing to ask. It's a slower way to lead in the short term and a much faster one over a period of years.

This connects directly back to the no-one-on-ones policy and the flat structure: all three habits are aimed at the same target, which is reducing how much of the company's judgment lives only inside one person's head. Explaining reasoning, keeping the group flat, and skipping private meetings are not three separate quirks. They're one strategy, applied three different ways.

It also raises the bar on the leader doing the explaining. Narrating your reasoning constantly means that reasoning has to actually hold up in front of an audience, repeatedly, rather than living safely behind a closed office door. That exposure is arguably its own quiet form of discipline.

Fast fact: Coworkers describe Huang narrating his reasoning out loud in nearly every meeting — not just announcing decisions, but showing the thinking behind them.

Betting on Something Before It's Obvious

13

NVIDIA's central technology, the graphics processing unit, was built originally for rendering video game graphics. For most of the company's early history, that was its entire market. The decision to keep investing in the same underlying chip architecture for general, non-graphics computation — years before anyone called it "AI infrastructure" — was a long, unglamorous bet that did not pay off quickly.

Huang has summarized his view of vision with a line that cuts against a lot of typical startup rhetoric: vision without execution is hallucination. A bold prediction about the future is worthless on its own. What matters is whether a person or company can actually do the unglamorous, repeated work of building toward that prediction for years before anyone else believes it's coming.

The pattern worth noticing is patience paired with narrowness. NVIDIA didn't chase every emerging trend in computing. It kept refining one core capability — parallel processing — and waited for the world's computing needs to bend toward it. That is a very different posture from constantly pivoting toward whatever looks exciting this year, and it's a much harder one to sustain, because it offers little validation along the way.

This is easy to admire in hindsight and much harder to practice in real time, because a long, unproven bet looks identical to a mistake for most of its lifespan. The real difference between the two, from the inside, is whether you're still willing to keep funding it years before the market agrees with you.

Vision without execution is hallucination.

JENSEN HUANG

The Steady Presence at Home

14

Huang met Lori Huang, his wife, while they were both engineering students at Oregon State University. They have remained married since, raising two children together, and Lori is frequently described by people close to the family as a grounding, steady presence against the intensity of Huang's work life. Public detail about their home life is intentionally limited — Huang has kept that part of his life largely out of interviews.

What's notable, given how much of this account focuses on relentless work, is that the marriage itself predates any of NVIDIA's success by years. It formed during the studying, unglamorous phase of Huang's life, not after wealth or recognition arrived. That ordering matters: it suggests a relationship built on shared effort during uncertainty, rather than one that arrived as a reward for it.

It would be dishonest to over-romanticize the balance here. Huang himself has acknowledged that work and rest blur together for him more than most people would find sustainable. The more accurate lesson isn't "he has perfect balance." It's that even inside a genuinely all-consuming career, he has maintained one long-term, stable relationship as a fixed point, rather than letting every part of his life stay in flux at once.

It's a useful counterweight to the rest of this account, too. A story built entirely around relentless intensity risks implying that everything else in a person's life is optional. Huang's own history suggests otherwise: the marriage held steady through the same thirty years of uncertainty as the company did.

Fast fact: Huang met Lori, his future wife, while both were engineering students at Oregon State — years before NVIDIA existed.

Run, Don't Walk — But Expect to Stumble

15

In a commencement speech, Huang told graduates to run, not walk, toward the opportunities ahead of them — while being explicit that they should also expect to fail along the way, repeatedly, and that this was not a warning sign to slow down. The pairing of those two instructions is the more interesting part than either one alone.

Most advice to "move fast" implicitly promises that speed and safety can coexist if you're careful enough. Huang's version doesn't make that promise. He treats stumbling as the expected cost of moving quickly, not as evidence that you moved incorrectly. This reframes failure from a signal to stop into a signal that you were actually moving at a realistic pace.

His approach to mentoring reflects the same idea at a smaller scale. Rather than shielding people from difficulty or quietly moving them aside after a mistake, he tends to keep investing in them through it — "torturing them into greatness," in his own phrase, rather than giving up early. The mentorship model isn't gentle, but it is patient, and those two things are not the same.

The instruction only holds together, though, alongside its second half. Telling people to move fast while punishing every stumble just produces fear. Telling people to move fast while treating stumbles as an expected cost produces something closer to actual courage.

Fast fact: In a commencement address, Huang told graduates to run, not walk, toward opportunity — and to expect to stumble along the way.

The Pattern Underneath All of It

16

Pull back from the individual stories and a single pattern repeats: discomfort gets treated as information, not as a problem to eliminate. The Kentucky boarding school, the Denny's shifts, the thirty days of cash, the flawed chip, the public correction of mistakes — in every case, the response was to stay in the discomfort long enough to learn something from it, rather than to move quickly toward relief.

This is a meaningfully different operating principle than "work hard" or "never give up," which are too vague to act on. The more precise version is: treat hardship as a source of calibration, disclose problems honestly and early even when it's costly to do so, and keep the flow of information as undistorted as possible between the people who need it.

None of this required unusual talent to start. It required a willingness to stay uncomfortable on purpose, in small doses, repeatedly, for a very long time. That is the actual throughline connecting a nine-year-old cleaning toilets in Kentucky to a CEO who still calls himself thirty days from failure, three decades and a trillion-dollar valuation later.

It's also, notably, a pattern rather than a personality trait — which matters, because patterns can be practiced deliberately, while traits get treated as fixed. Nothing about staying in discomfort long enough to learn from it requires being a particular kind of person from birth.

Fast fact: The same pattern — discomfort treated as information — runs from a Kentucky boarding school all the way to a trillion-dollar boardroom.

Ten Principles, Stated Plainly

17

1. **Start the day on your terms.** Move your body and set your priorities before the world sets them for you.
2. **Treat hardship as calibration.** Difficulty survived recalibrates what you believe you can handle next.
3. **Disclose bad news early and honestly.** The uncomfortable truth, told first, is what earns you the second chance.
4. **Keep information undistorted.** Every layer between a fact and a decision-maker is a chance for that fact to get softened or lost.
5. **Reason out loud.** Show your thinking, not just your conclusions, so others can learn the pattern, not just the answer.
6. **Correct mistakes where others can learn from them.** One person's error, handled openly, can prevent ten repeats of it.
7. **Be patient with people, not with problems.** Give people time to grow; don't give unresolved problems time to fester.
8. **Keep learning on purpose.** Protect real time for reading and staying current, deliberately, not as an afterthought.
9. **Pair vision with unglamorous execution.** A prediction about the future is worth nothing without years of boring work behind it.
10. **Keep one thing stable.** Even amid total professional intensity, anchor to at least one steady, long-term relationship.

What Not to Copy

18

It's worth ending on a note of caution rather than pure admiration, because copying this story uncritically would miss its own lesson. Huang's seven-day work weeks are described, by his own words, as coming from anxiety rather than balance. That is not a template to aim for. A habit that works for one person, forged by one specific set of early experiences, does not automatically transfer well to someone else's life or nervous system.

The useful parts of this story are the transferable principles: honest disclosure, undistorted information, treating discomfort as data, patience with people. The less useful part is the literal schedule. Sustained anxiety is not a productivity hack, even when it happens to sit next to genuine achievement. Anyone drawing lessons from this account should feel free to take the discipline and leave the sleep deprivation.

The most accurate summary of Huang's story isn't "work constantly and you'll succeed." It's closer to: build a habit of staying honest under pressure, keep your information clean, and don't let comfort talk you out of the work that actually matters. Those habits can be practiced at a sustainable pace. They don't require recreating anyone's anxiety to be effective.

The distinction worth holding onto is between a habit's output and a habit's cost. The output — honesty, clarity, sustained attention — is worth wanting. The cost, in this specific case, was thirty years of never fully switching off. Those two things can, and should, be separated when deciding what to actually adopt.

Fast fact: Huang has openly called his seven-day-a-week schedule a product of anxiety, not balance — not a template he says people should copy.

Where to Start This Week

19

None of the principles in this account require a company, a title, or a technology bet to begin practicing. They can start with something as small as one meeting this week. Try explaining your reasoning out loud before announcing your decision, and notice whether the people around you engage differently. Try disclosing one piece of bad news earlier than feels comfortable, before it's forced out of you, and notice what it does to trust rather than to your reputation.

Pick one recurring source of low-grade discomfort in your work — a task, a conversation, a piece of feedback you've been avoiding — and treat it deliberately as information rather than as a threat to manage away. That single shift, repeated often enough, is closer to the real substance of this story than any fact about revenue or chips.

Jensen Huang's story is not really about being a founder. It's about what a person does with discomfort, honesty, and attention over a very long stretch of ordinary days. That part is available to anyone, starting today, regardless of what they build.

If only one habit from this account is worth keeping, let it be the earliest one: tell the truth about a problem before you're forced to, even when it costs you something in the moment. Everything else in this story — the flat structure, the public corrections, the long bets — is really just that one habit, applied at a larger and larger scale.

"Suffering is part of the journey. You will appreciate it so much more when things do go well."