GitHub

HACKATHON

TIPS AND TRICKS

SO YOU WANT TO RUN A HACKATHON?

There are a lot of different types of hackathons you can run. Some are simply ideation sessions, others focus on the first stages of solving a problem, and others still are designed to teach people certain concepts, like human-centered design, particular tech stacks like JavaScript, or certain products like GitHub Actions.

Don’t believe the Hollywood movies! In general, a hackathon usually has nothing to do with “breaking into things”. Hacking refers to taking a problem or issue, breaking it down into its root cause, and finding a viable solution.

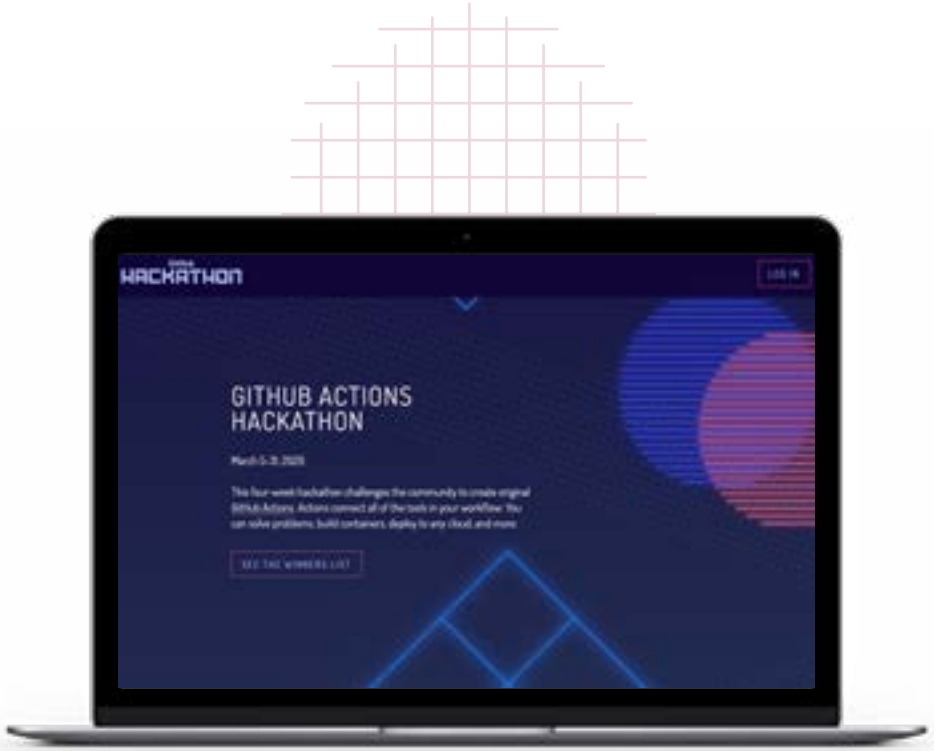
A hackathon is a short competition where people work together in teams to solve problems and challenges by coming up with solutions and ideas.

WHAT IS A GITHUB HACKATHON?

GitHub Hackathons are focused on technology solutions usually underpinned by code. Hackathons are focussed on identifying problems and coming up with creative solutions, presenting a proof of concept or Minimum Viable Product (MVP) for that solution.

Rather than creating brand new ideas, your hackathon could also be a coding challenge, where you post a problem and developers must use code to solve that problem. The problem could be “build a GitHub Action”, or it could be “solve these issues in this example program”. Coding challenges are great to learn a new technology space while working together in teams or even as an individual.

Hackathons focus on identifying problems, coming up with creative solutions then presenting a proof of concept for that solution.



The GitHub Hackathon website contains all information for hackathons.
https://githubhackathon.com/

GITHUB HACKATHON CRITERIA

To organize an event that qualifies as an official GitHub Hackathon, these are the things you will need to implement:

- The hackathon projects and ideas MUST be available in a public GitHub repository and published with a recognized open source license. When you create your code repository in GitHub, we help you pick a license or see https://choosealicense.com/ for more information.
- The hackathon MUST enforce a code of conduct to help ensure the event is inclusive and welcoming to all. The type of code of conduct varies depending on if your hackathon is in-person or online. For an in-person event, something like the GitHub Event Code of Conduct is a great start. For any open source repo, you'll need a Code of Conduct, which can be found on the Contributor Covenant.

GitHub Event Code of Conduct and Contributor Covenant
https://help.github.com/en/github/site-policy/github-event-code-of-conduct

WHY WOULD DEVELOPERS COME TO YOUR HACKATHON?

Developers love working on hackathons for a number of reasons. When it comes to coding, hackathons are a perfect way to practice skills and network with people. Here's some of the reasons why developers go to hackathons:

- Work on problems and solutions they care about
- Build their CV/GitHub profile through creation of products
- Learn and develop skills
- Practice pitching
- Network with people in an industry
- Opportunity to work on/with great software, technology, datasets
- Of course, people always appreciate any freebies and SWAG!

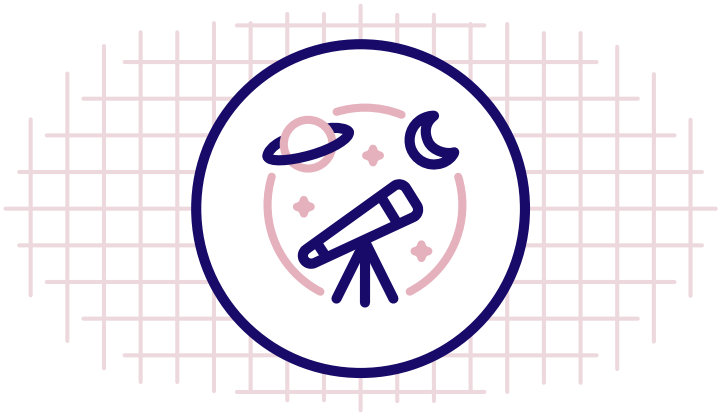
When creating a hackathon, you should think about what can you provide people as incentives to attend. How are you going to stand out and convince developers your event is a great way to spend their free time?

WHAT IS THE THEME FOR YOUR HACKATHON?

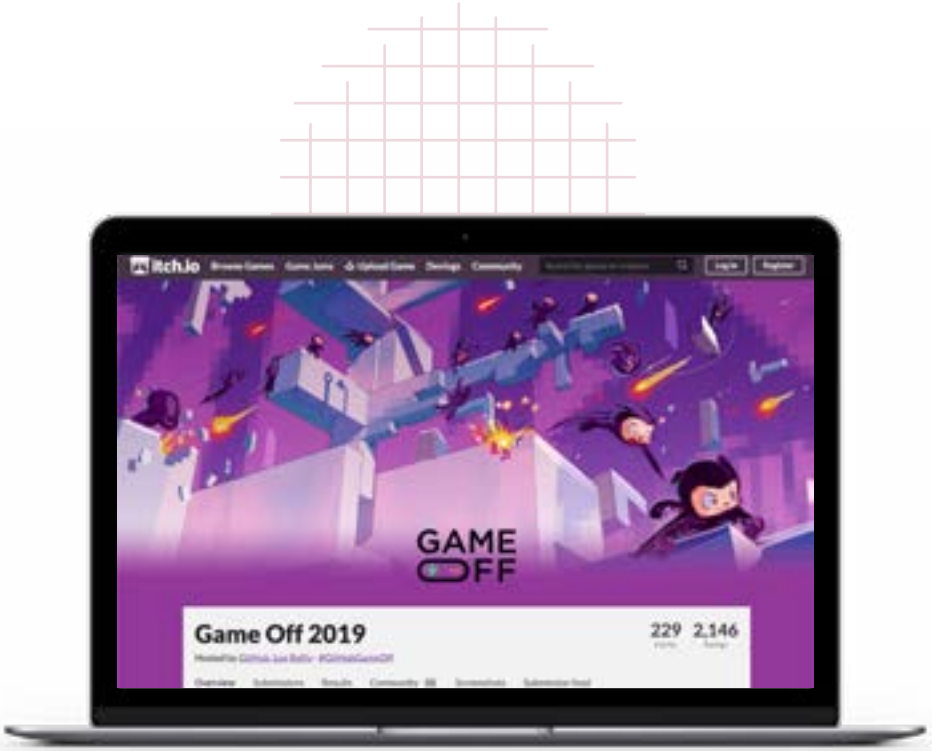
Your theme could be very specific (ie. how to use AI to solve traffic congestion in a city) or it could be something more open (ie. how to unlock a city's potential and increase urban mobility?) The less specific themes are often better as it opens up to outside the box thinking. The more narrow the topic, the more similar ideas you'll get. If you want lots of different ideas for various things, then open up the theme.

You don't need to have a theme. Depending on your goals, simply having a "build something, anything, in 24 hours, and present the solution for review" can be a good way of encourage networking and team building. These hackathons often have a "it has to work" theme rather than a specific problem to solve. This leaves the hackathon completely open to the participants to define their own problems and suggest a solution for it.

The less specific themes are often better as it opens up to outside the box thinking. The more narrow the topic, the more similar ideas you'll get. If you want lots of different ideas for various things, then open up the theme.



Whatever theme you choose, whatever data sets, or technology you decide to have, it's important to remember you'll need mentors and experts there to help.



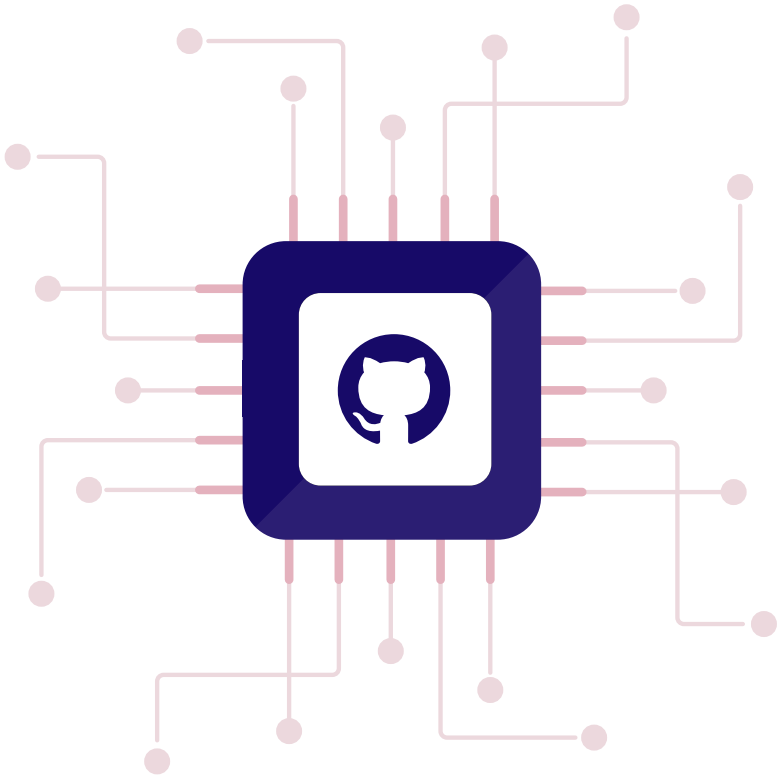
GitHub Game Off's only guidance is that participants have to build a game.
https://itch.io/jam/game-off-2019

It can be really great for encouraging new startup ideas, business ideas, and when trying to recruit people for innovative thinking.

Other times you may want a theme, but still remember to keep it relatively broad. Themes like "MedTech", "aged care", "transportation", or similar are great because they are very broad and can be taken in a lot of different contexts. Another example is the **GitHub Game Off** where the only guidance is that participants have to build a game. Other times you may want to narrow it down a little further. Slightly more narrow themes such as "How can we unlock a city's potential?", "how can we use machine learning in MedTech", or "how can we use data X for the purpose of X industry?". These are good because they are broad enough for people to interpret them, and narrow enough that desired outcomes are achieved.

Avoid having a hackathon around something very narrow. For example “build a real time strategy video game to inspire kids to do their homework”. This would ensure all the ideas are very similar, which is not what you want.

Whatever theme you choose, whatever data sets, or technology you decide to have, it’s important to remember you’ll need mentors and experts there to help. For example, if you are using a data set that shows information around traffic congestion in the city, it’s useful to have someone from the transport industry to help explain the data. If you’re designing a hackathon to develop on a new blockchain platform, ensure the experts are available to help with that platform. Otherwise you’ll find participants will have questions that can’t be answered quickly and instead might give up and go home.



Not sure where to start as a developer? Check our handy guides and reference documentation you can use to start building.
https://developer.github.com/

IS THE EVENT IN-PERSON OR ONLINE?

You'll need to decide if this event is virtual (online) or physical (in-person). There are pros and cons to each type of event. Let's take a look at what those are.

ONLINE EVENT

ADVANTAGES

- Not limited by region, therefore you can open it up to have more people participate
- Barrier to entry is low; anyone with an internet connection can participate
- Better for long form events; ie. a number of weeks
- Easily recorded as you go
- Heaps of online tools to help you

DISADVANTAGES

- Much harder to create "connection" and "atmosphere" in a digital only world
- Participant networking is more difficult
- Harder to coordinate people into teams
- Harder to know if people need more help and to build connections with your mentors

IN-PERSON EVENT

ADVANTAGES

- Build a great culture amongst people within the room
- Motivate the participants with "hype" and "atmosphere"
- People can easily learn from one another if in the same room
- Allows for serendipitous encounters
- Useful for short form events (ie. 5 hours up to a couple of days)
- Easy to hold a "science-fair" style demo pit of all the solutions/ideas
- People are more likely to participate while in the room due to "seeing" other people working
- Help builds a collaborative culture

DISADVANTAGES

- Venue hire, food, and other physical location costs will need to be covered by fundraising
- Need a good MC to tie everything together and keep the energy going and have in-person mentors available

WHAT IS THE AGENDA OF THE HACKATHON?

Now you have decided on online or in-person you also need to decide the type of format you want.

These are some of the things you can consider implementing into your hackathon plan.

KICK OFF

The opening ceremony is your hype train. This is your opportunity to present to participants, sponsors, VIPs, and supporters about the hackathon. In the opening ceremony you can consider adding:

- Welcome from a sponsor or VIP
- Information around what the problem is or why you decided to run a hackathon
- Information from sponsors
- Tips and tricks for a successful hackathon from a participant's point of view
- Explain how the results will be judged (ie. presenting, judging, voting etc.)
- Any team formation exercises if needed

CHECK IN SESSIONS

Regular review periods where you focus attendees attention on what is happening and where people are at. This will help keep participants on track but sometimes is also an opportunity for teams to join resources or for people to swap teams if another solution sounds like something they would have more fun implementing.

WORKSHOPS

In a longer running hackathon, you might also hold workshops. Giving help on pitching, coding, design thinking, open source, maintaining projects, problem solving etc. All these workshops will not only upskill participants but will also help them develop good ideas and contribute to a successful hackathon.

CLOSING CEREMONY

Whether online or in person, it's great to close out the hackathon:

- Presentation of ideas
- Judging of ideas
- Presentation of prizes
- Recap hackathon
- Thank yous
- Next steps (see below)

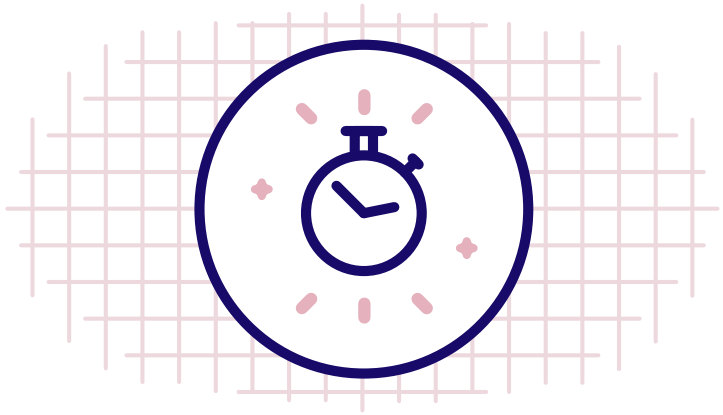
Other things you might like to consider depending on budget and goals.

- Food and drink breaks: mix groups up and allow them to network with a wider group of people
- Yoga sessions (great for 24 hour+ hackathons)
- Mini competitions
- Dance Central
- Social media
- Best memes
- Massages
- Video sessions: vox pops with participants, sponsors, and organizers
- LEGO Serious Play

HOW LONG SHOULD THE HACKATHON BE?

Most physical, in-person hackathons will be run over one to three days. This is due to the intense nature of physical events and the required energy from both participants and organizers.

Virtual coding hackathons can last anywhere between a couple of days, and a few weeks. Think about what you're trying to achieve and ensure you give participants adequate time to build out their ideas.



WHAT TYPES OF PEOPLE WILL THE HACKATHON SUIT?

Even though GitHub Hackathons are about solving problems with technology, the most successful projects always have a diverse set of people involved. Even when it comes to coding, you don't want all the same people in one team. For example, if you have all front-end developers, you'll get lots of nice websites, but no back-end functionality. If it's all coders then the design might be terrible, the documentation or video not understandable or the business idea not well thought through.

Most hackathon goers would be aware of the three main roles in a hacakthon. We've called these: **coder**, **innovator**, and **designer**. These three terms refer to the different roles within a hackathon team (even a team of one). If you have the right mix of these three personas, you'll have the perfect dream team!

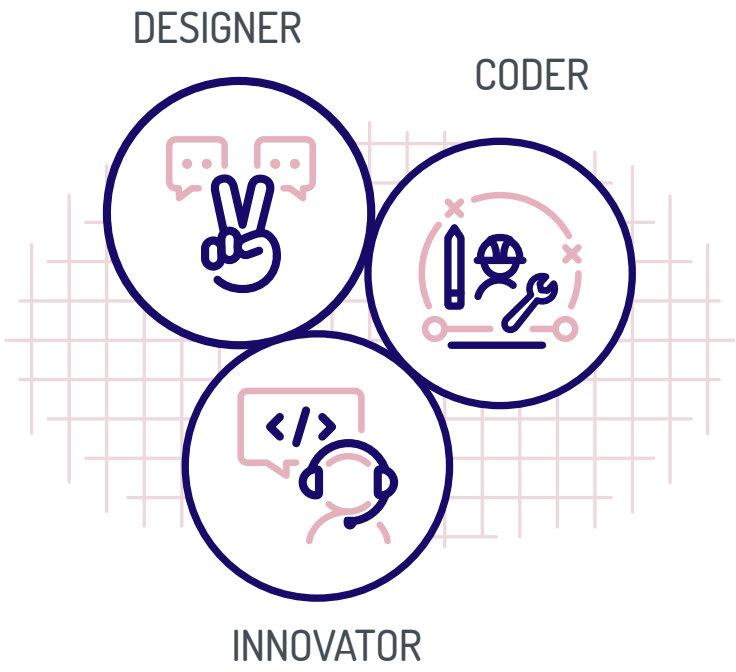
SO WHO ARE THESE PEOPLE AND WHAT'S THEIR ROLES?

CODER: your techies. Engineers and developers are your coders. They are the dev gurus and usually have good knowledge of many different tech stacks. A more technical focused hackathon might have multiple coders in the group, just like a product or service will have multiple developers working on it together.

INNOVATOR: your business and marketing people. They are the ones who go out and validate the ideas, understand the problem, research the market, and usually they'll be the one presenting the final product.

DESIGNER: your creative type. Every project needs great design, from beautiful UI to UX. These people focus on the product itself and how it looks and feels to the customer.

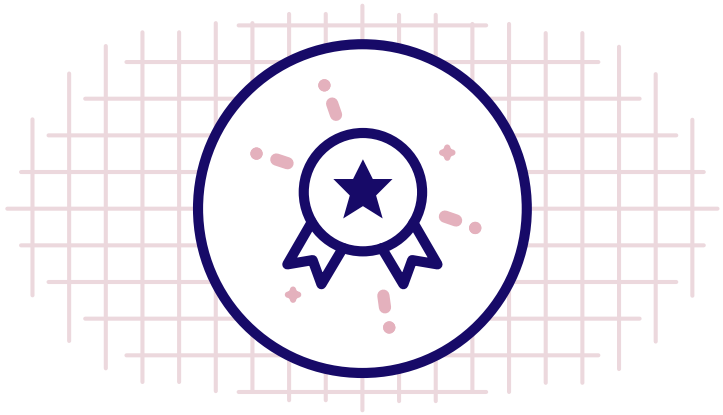
Any one person could have multiple skill sets that cover multiple roles above. For a hackathon however where there is a short timeframe required, it's best to pick one role and focus your efforts there. It can also be more beneficial to the team as roles are clearly defined and the team will be able to deliver something faster.



HOW SHOULD WE 'JUDGE' THE IDEAS PRESENTED?

First thing to note, not every hackathon needs to have judging. For example, the GitHub Actions Hackathon is more a "challenge" where participants completed the steps outlined, and there were no "winners"—simply participating and delivering a solution that met the criteria was enough to qualify for a reward.

But a little light hearted competition can go a long way to motivate people. When it comes to coding, here's some of the areas to think about judging.



CONSIDERATIONS

APPROACHABLE: Is the code easy to understand and contribute to?

SOLUTION USABILITY: is the end result easily used? It is utilising current tech stacks and practices?

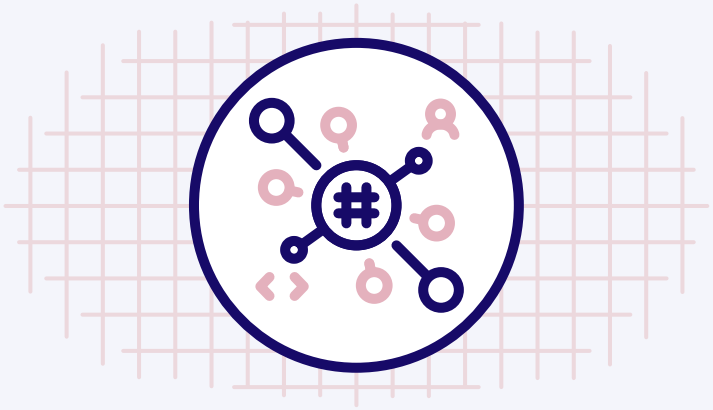
DOCUMENTATION: how well has the team documented their solution? Is it easy to understand? Is there a "read me" file to get people started?

SECURITY: is the code secure? Has the team thought about vulnerabilities and/or denial of service issues?

WHAT WILL YOU DO AT THE END OF THE HACKATHON?

So all the fun is over, now what? There's a number of options for what to do after a hackathon.

What you choose will depend on the reason for running the hackathon, what resources you have available, and what you're able to do. Here's some things you can do at the end of the hackathon:



PRESENT IT

Showcase the ideas in a public forum. Create a video, write a blog post, or have them on your website. Participants will be able to see the projects they created and tell their friends. Others in the community can see how successful your hackathon was.

SHARE IT

Your hackathon was successful and new ideas were created, and problems solved. Why not celebrate that? Post about it on social media, share some highlights and more.

Encourage participants to share on their channels, and to put their accolades on places like LinkedIn. For an example, check out the GitHub Actions Hackathon blog post.

SHOWCASE IT

Encourage participants to add their hackathon creations to their own LinkedIn and GitHub profiles. Using the "pin project" on each individual's GitHub profile is a great way to showcase what's been built.

INCUBATE IDEAS

Some people in the hackathon might like to continue to build some of the ideas. If you have people who can help your participants take their ideas to the next level, do it!

IMPLEMENT CONCEPTS

Encourage your participants who attended the hackathon to understand the concepts learned during the hackathon and get them to apply them to everyday work life. Innovation, creativity, problem solving, critical thinking, teamwork, communication, and customer-centric thinking, should all become part of the way you work.

DO ANOTHER HACKATHON!

Liked the hackathon? Then run another one! Some hackathons run annual, half-yearly, and even mini-monthly hackathons.

Explore upcoming GitHub Hackathons at:

https://githubhackathon.com

GitHub

HACKATHON

QUESTIONS ABOUT CREATING YOUR HACKATHON? WE'RE HERE TO HELP.

GITHUBHACKATHON.COM