Course Companion

GAC Computing

Every topic handout in one volume

全部专题讲义合集

exercises.lol

Contents 目录

---

1. GAC005 Computing I: Introduction to Computing for Academic Study 2

2. GAC011 Computing II: Web Development 7

3. GAC017 Computing III: Data Science and Web Apps 10

GAC005 Computing I: Introduction to Computing for Academic Study

GAC Computing

What this module is, and how it is marked

GAC005 is not a programming course. It is about using a computer to do academic work well: finding sources you can trust, writing with other people, running a small project, and presenting the result.

Everything is assessed on what you produce. A typical set is a **search and writing task**, a **group project** 小组项目 with a presentation, a **video** 视频, and ongoing **coursework** 平时作业. There is no written examination, so the files you submit are the whole mark.

- Work is judged on the artefact and on the process. Keep drafts and notes; a group project is marked partly on your contribution to it.
- △ **Academic integrity** 学术诚信 applies to media as much as to text. An image, a chart or a clip taken from the web needs a source, and a licence that allows the use.

Search literacy

Anyone can find a page. The skill being assessed is finding a source you can defend.

- Search with **keywords** 关键词, not sentences. Pick the nouns that must appear in a good answer and drop everything else.
- **Operators** 运算符 narrow a search: quotation marks force an exact phrase, a minus sign excludes a word, and `site:` restricts results to one domain.
- Judge a source before you use it. **Authority** 权威性 asks who wrote it and what they know; **currency** 时效性 asks how old it is; **bias** 偏见 asks what the author wants you to think.
- A **primary source** 一手资料 is original material, such as the report of an experiment. A **secondary source** 二手资料 describes or interprets it, such as a news article about that report.
- **Search results are ranked, not verified** – position on the page says nothing about accuracy.

> **Worked example.** The question is whether electric cars produce less carbon dioxide overall than petrol cars. A search for the sentence returns opinion pages. A search for the keywords *lifecycle emissions electric vehicle comparison* returns studies, and adding `site:gov` narrows it to government reports whose authors and methods are stated. The second search is not cleverer, it is more specific about what a usable answer looks like.

```
electric vehicle "lifetime emissions" -advert site:gov.cn
```

keywords: nouns, not a sentence — quotation marks: exact phrase — minus: exclude — site: one domain

then judge each result: authority, currency, bias; rank is not truth

Each part of the query narrows the results; judging them is still your job.

Collaborative writing

Writing together is a normal way to work, and the module assesses how you do it.

- A **cloud document** 云端文档 holds one live copy that everyone edits, which removes the "final_v3_really_final" problem entirely.
- **Version history** 版本历史 records who changed what and when. It protects you: your contribution is visible even when a file looks like one voice.
- **Comments** 批注 propose a change without making it; **suggestions** 建议模式 make a change the owner can accept or reject. Use them instead of overwriting someone's paragraph.
- Agree the **structure** 结构 before anyone writes. Splitting a document by section works; splitting it by sentence does not.
- Keep one **style** 文体: decide on person, tense and spelling at the start, and give one person the final read.

Project management

A group project fails on organisation far more often than on ability.

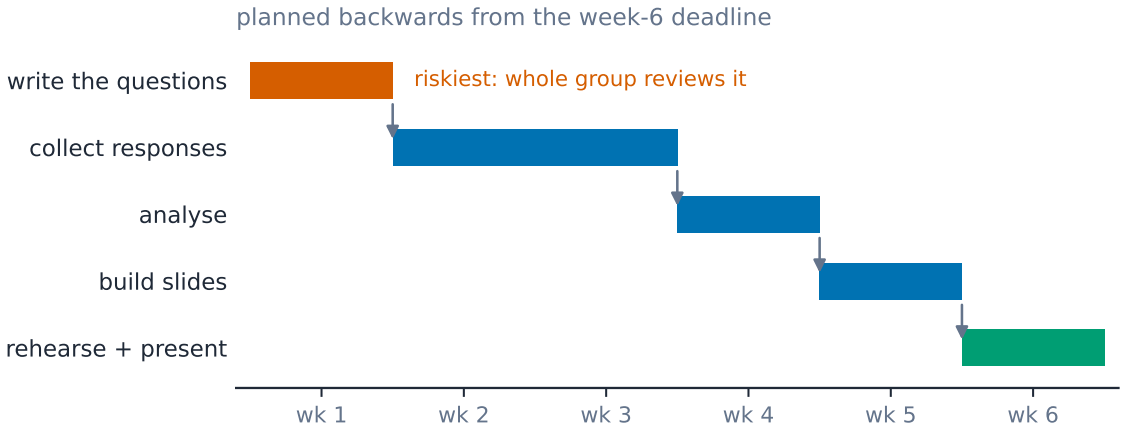


A team planning a project: clear tasks, owners and deadlines keep shared work on track

- A **scope** 范围 statement says what the project will and will not do. Most overrun starts as an unwritten extra.

- **Milestones** 里程碑 are checkpoints with dates. They convert "we have three weeks" into something you can be behind on early enough to fix.
- Assign **roles** 分工 to people, not tasks to the group. A task owned by everyone is owned by nobody.
- A **team contract** 团队协议 records how the group will meet, decide, and handle a member who does not deliver. Writing it while everyone is friendly is the point.
- Track **dependencies** 依赖关系: the slide deck cannot be finished before the data is collected, so the data has the earlier deadline.

> **Worked example.** A four-person project to survey and present student transport habits, due in week 6. Working backwards: rehearse in week 6, build slides in week 5, analyse in week 4, collect responses in weeks 2 to 3, write the questions in week 1. The questions are the riskiest item because everything downstream depends on them, so they get reviewed by the whole group and nothing else in week 1 does.



The transport survey planned backwards from week 6: every task waits for the one above it

Creating effective presentations

A presentation is judged on whether the audience understands, not on how much is on screen.

- One **message** 信息 per slide. If a slide needs two sentences to summarise, it is two slides.
- Slides support the speaker. **Bullet points** 要点 are prompts, not a script, and reading them aloud is the most common way to lose marks.
- **Contrast** 对比度 and size decide whether the back row can read it: dark text on light, at least 24 point, and a **sans-serif** 无衬线 typeface.
- Use a **chart** 图表 when the shape of the data is the point, and a **table** 表格 when exact values are. A chart needs axis labels and units, always.

- Cite sources on the slide where the material appears, and keep a full **reference list** 参考文献列表 at the end.

Enhancing your presentation

Once the content is right, delivery is what remains.

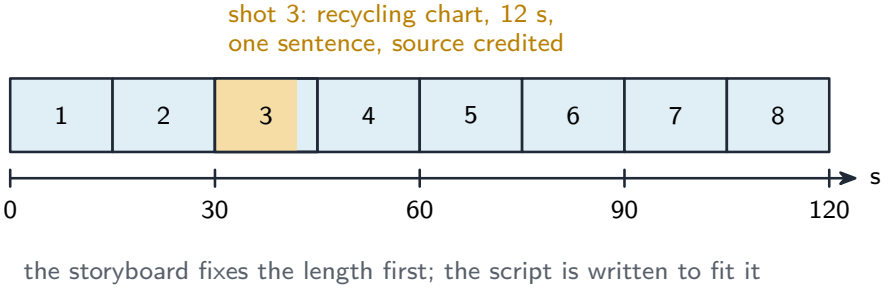
- Rehearse **aloud** 出声, standing, with a timer. Reading silently hides every sentence you cannot actually say.
- **Signposting** 提示语 tells the audience where they are: "first", "the reason for this", "to sum up". In a second language it does more work than any slide animation.
- **Eye contact** 眼神交流 and pace matter more than vocabulary. Slower is almost always better.
- Prepare for **questions** 提问 by listing the three you would least like to be asked, and answering them.
- **Animation** 动画 has one legitimate use: revealing a complex diagram one part at a time. Everything else distracts.

The making of a video

A short video is assessed like any other academic product: planned, sourced and edited.

- A **storyboard** 分镜脚本 plans each shot before recording: what is on screen, what is said, how long it lasts. It is the difference between an hour of editing and five.
- A **script** 脚本 keeps a two-minute video to two minutes. Speaking unscripted takes three times as long and says less.
- **Audio quality** 音质 matters more than picture quality. A quiet room beats an expensive camera, every time.
- Use **licensed** 授权 material only: images and music released under a licence that permits reuse, credited on screen or in the description.
- **Export** 导出 to a common format at a sensible size, and check that it plays on a machine that is not yours before you submit it.

> **Worked example.** A two-minute explainer on plastic recycling. The storyboard has eight shots of about fifteen seconds. Shot 3 is a chart of what a city actually recycles, on screen for twelve seconds with one spoken sentence, credited to the source in the corner. Because the storyboard fixed the length first, the script was written to fit the time, rather than the video being cut to fit a script that was already too long.



the storyboard fixes the length first; the script is written to fit it

Eight shots fix two minutes before a word of script is written.

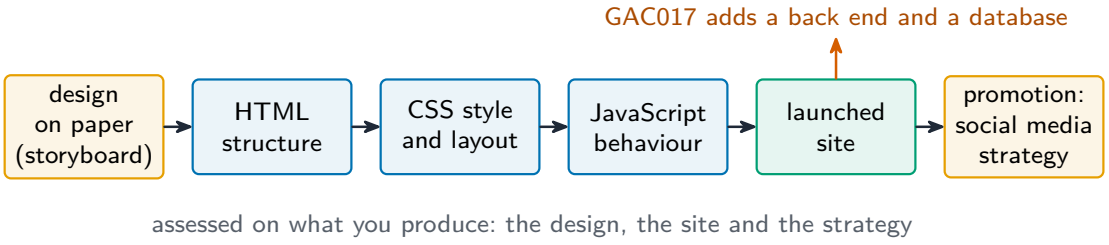
GAC011 Computing II: Web Development

GAC Computing

What this module is, and how it is marked

GAC011 is where you build and publish a real website. Six units take you from a design on paper through HTML, CSS and JavaScript to a launched site and a plan for promoting it.

Everything is assessed on **what you produce**: a **storyboard** 分镜脚本 and design, the site itself, and a **social media strategy** 社交媒体策略, plus coursework. There is no written exam.



Six units from a paper design to a launched, promoted site

- **Design counts as much as code.** The module's outcomes name aesthetics, layout, navigation and usability alongside the scripting.
- △ The site you design here is the thing GAC017 gives a back end and a database. A weak plan now costs time in the next module.

You can run every example in this module in the site's own code playground, and the HTML, CSS and JavaScript references there go deeper than these notes do.

Designing a website

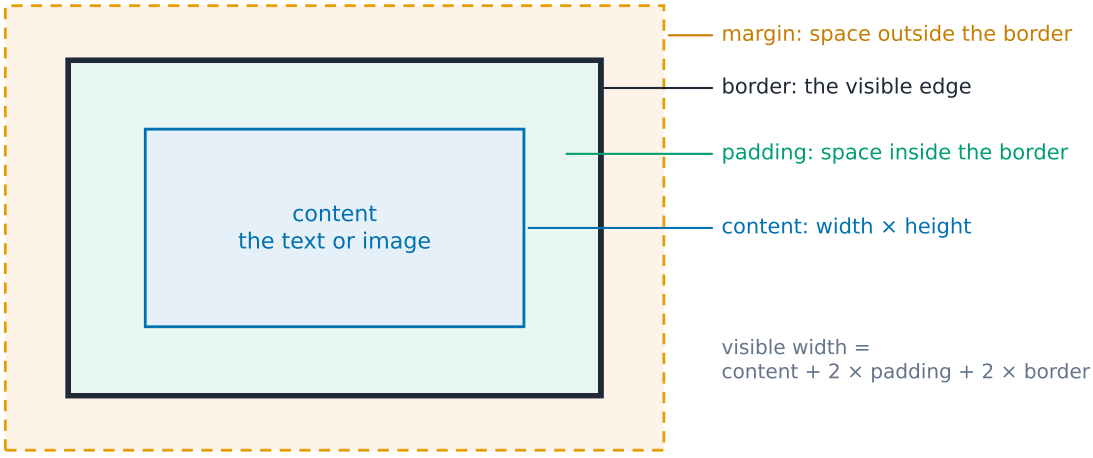
- Start on paper. A **wireframe** 线框图 shows structure without colour, which is the only way to argue about structure.
- **Information architecture** 信息架构 is what goes where: the pages, and the path a visitor takes between them.
- **Visual hierarchy** 视觉层次 tells the eye what matters – through size, weight, spacing and contrast, not through decoration.
- Design for the **audience**, and say who they are before you choose anything.

Creating a website using HTML

- **HTML** 超文本标记语言 marks up meaning, not appearance. A heading is `<h1>` because it *is* a heading.
- An **element** 元素 has an opening tag, content and a closing tag; an **attribute** 属性 goes in the opening tag.
- **Semantic elements** 语义化元素 – `<header>`, `<nav>`, `<main>`, `<article>`, `<footer>` – tell a browser, a search engine and a screen reader what a region is.
- Every image needs **alt text** 替代文本. It is what a blind visitor hears and what a search engine reads.

Using CSS to enhance your website

- **CSS** 层叠样式表 controls appearance, and keeping it out of the HTML is the whole point of the separation.
- A **selector** 选择器 chooses elements; a **declaration** 声明 sets a property to a value.
- The **box model** 盒模型 explains layout: content, then padding, then border, then margin.
- **Responsive design** 响应式设计 makes one page work on a phone and a laptop, usually with flexible widths and a **media query** 媒体查询.



The box model from the inside out: content, padding, border, margin

Worked example. A card that is too wide on a phone.

The fix is not a fixed pixel width. Set `max-width` so the card never exceeds the screen, let it shrink below that, and add a media query only for the change that a smaller screen genuinely needs, such as stacking two columns into one.

Designing the flexible case first, and treating the media query as the exception, is what "responsive" means.

Adding interactivity with JavaScript

- **JavaScript** runs in the browser and changes the page after it has loaded.
- The **DOM** 文档对象模型 is the page as a tree of objects your code can read and change.
- An **event** 事件 – a click, a submit, a key press – triggers a **function** 函数 you wrote.
- Keep it small at this level: validate a form, toggle a menu, filter a list. A site that fails when JavaScript fails is a design error, not a coding one.

Launching your website

- **Hosting** 托管 puts your files on a server; a **domain name** 域名 points visitors at them.
- **Test before you launch**: every link, on a phone, on a browser that is not yours.
- **Accessibility** 无障碍 is not optional – contrast, alt text, keyboard navigation, and text that resizes.
- △ Check what you are allowed to publish. An image or a font on your site needs a licence that permits it, exactly as a quotation in an essay needs a citation.

Social media marketing

- Decide the **audience** and the **objective** before choosing a platform. "Be on social media" is not an objective.
- A **content plan** 内容计划 says what you will post, how often, and who writes it.
- **Metrics** 指标 must connect to the objective: reach, engagement, and click-through to your site are different things, and only one of them is traffic.
- Measure, then change one thing. Changing everything at once teaches you nothing about which change worked.

GAC017 Computing III: Data Science and Web Apps

GAC Computing

What this module is, and how it is marked

GAC017 is the Level III computing module: what sits behind a website. Six units cover back-end JavaScript, databases, SQL, connecting the two, and a first look at data science.

Assessment is entirely practical – a **database** 数据库 you design and build, a **web app** 网络应用 that talks to it, a **data analysis project** 数据分析项目, and coursework – usually spread across most of the semester rather than concentrated at the end.

- The work is judged on whether it **runs and answers a question**, not on how much code there is.
- △ Design the database before you write a query. Almost every problem later is a table problem wearing a query's clothes.

Every SQL example here runs in the site's own code playground, and the SQL reference there is the fuller version of these notes.

JavaScript for a web app's back end

- The **front end** 前端 runs in the browser; the **back end** 后端 runs on a server and holds what the browser must not.
- A **server** 服务器 receives a **request** 请求 and returns a **response** 响应. That loop is the whole architecture.
- An **API** 应用程序接口 is the agreed shape of those requests and responses.
- △ Anything secret – a password, a key – belongs on the back end only. Code in a browser is readable by everyone who visits.

Back-end programming

- A **route** 路由 maps a URL to code that runs when that URL is requested.
- **Validate input** 校验输入 on the server. Checking in the browser is a convenience for honest users, not a defence.
- **Never build a query by joining strings** with user input. That is how **SQL injection** SQL 注入 happens, and it is the security failure this unit exists to prevent.
- Return an honest **status**: 200 for success, 400 for a bad request, 404 for something that is not there.

Introduction to databases

- A **relational database** 关系数据库 stores data in **tables** 表 of rows and columns.
- A **primary key** 主键 identifies a row uniquely. A **foreign key** 外键 points at another table's primary key, and that pointer is the relationship.
- **Normalisation** 规范化 removes duplicated data so one fact lives in one place.
- Design by asking what the **entities** 实体 are, then what connects them.



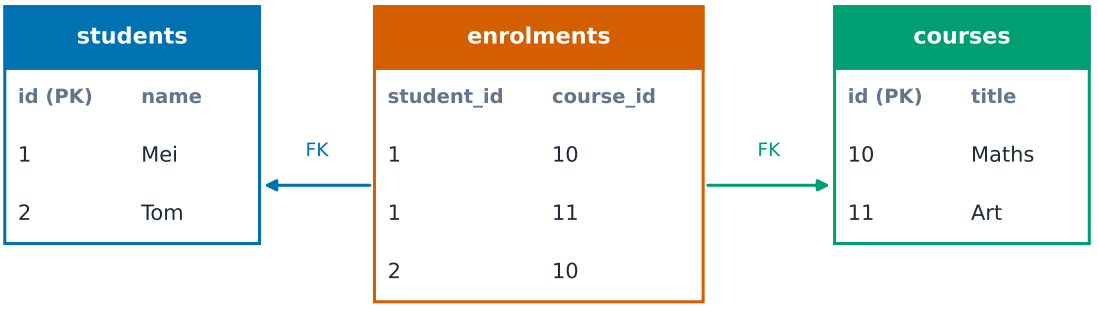
Server racks in a data centre: a web app's database lives on machines like these

> **Worked example.** A school wants to store students, courses and who takes what.

Two tables cannot do it: a student takes many courses and a course has many students. The many-to-many needs a third table – enrolments – whose rows are (student, course) pairs.

Recognising that this third table is required is the single most useful database idea in the module, and it is where most first designs go wrong.

exercises.lol exercises.com.cn



Mei takes two courses; Maths has two students. Each row of enrolments is one pair.

The many-to-many fix: an enrolments table of (student, course) pairs

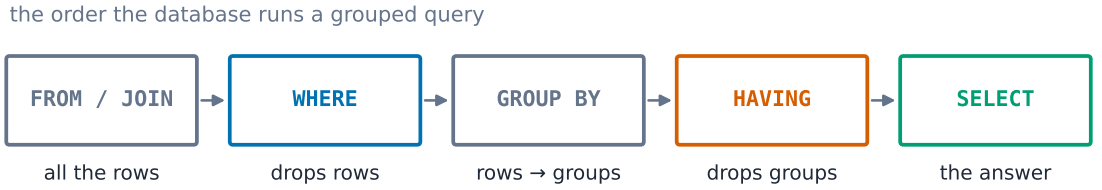
SQL for back-end programming

- **SQL** 结构化查询语言 asks a database questions. `SELECT … FROM … WHERE …` is the core.
- **JOIN** 连接 combines rows from two tables on a matching key.
- **GROUP BY** 分组 with `COUNT`, `SUM` or `AVG` answers "how many per…" questions.
- △ `WHERE` filters rows before grouping; `HAVING` filters groups after. Using the wrong one is the classic SQL error, and it usually returns a plausible wrong answer.

Worked example. Which courses have more than 20 students?

```
SELECT c.title, COUNT(*) AS students
FROM enrolments e
JOIN courses c ON c.id = e.course_id
GROUP BY c.id, c.title
HAVING COUNT(*) > 20;
```

The count is a property of the group, so the filter is `HAVING`. Grouping by `c.id` as well as the title keeps two courses that share a title as separate groups. Written with `WHERE` it does not run – and when a similar mistake does run, it silently answers a different question.



The order a grouped query runs in: WHERE before the groups exist, HAVING after

Connecting JavaScript with SQL

- The back end takes a request, runs a **parameterised query** 参数化查询, and returns the rows as data, usually **JSON** 数据交换格式.
- Parameterised means the values travel separately from the query text. That is what makes injection impossible rather than unlikely.
- Handle the empty case. A query that returns no rows is normal, and a page that breaks on it is not finished.

Data science

- **Data science** 数据科学 turns data into a decision, and most of the work is before the analysis: cleaning, joining, and checking what the data can support.
- **Descriptive statistics** 描述性统计 summarise; a **visualisation** 可视化 shows shape; neither proves a cause.
- **Correlation is not causation** 相关不等于因果 – the sentence every data project needs and most omit.
- State the **limitations** of your dataset. A project that names what its data cannot show scores above one that quietly overclaims.