# Information representation

A-Level Computer Science · Vocabulary in context

| Word | Example |
|---|---|
| **number system** /ˈnʌmbə ˈsɪstəm/ | Binary and denary are two kinds of **number system**. |
| **binary** /ˈbaɪnəri/ | In **binary**, the number 5 is written 101.<br>5 101 |
| **denary** /ˈdiːnəri/ | We count in **denary**, using the digits 0 to 9.<br>0 9 |
| **digit** /ˈdɪdʒɪt/ | In 352, the first **digit** is 3.<br>352 3 |
| **place value** /pleɪs ˈvæljuː/ | In 352, the **place value** of the 5 is ten.<br>352 5 |
| **byte** /baɪt/ | One **byte** can store one letter, such as A.<br>A |
| **bit** /bɪt/ | A **bit** is a single 0 or 1.<br>0 1 |
| **hexadecimal** /ˌheksəˈdesɪml/ | In **hexadecimal**, the number 255 is written FF.<br>255 FF |
| **nibble** /ˈnɪbl/ | A **nibble** is half a byte, or four bits. |
| **register width** /ˈredʒɪstə wɪtθ/ | A **register width** of 8 bits allows 256 different values.<br>8 256 |
| **register** /ˈredʒɪstə/ | The processor keeps the number it is using in a **register**. |
| **unsigned** /ʌnˈsaɪnd/ | An **unsigned** byte can store 0 to 255.<br>0 255 |
| **BCD** /ˌbiː siː ˈdiː/ | In **BCD**, each denary digit gets its own four bits. |
| **overflow** /ˌəʊvəˈfləʊ/ | Adding 200 and 100 in one byte causes **overflow**.<br>200 100 |



exercises.lol exercises.com.cn

| Word | Example |
| --- | --- |
| **most significant bit** /məʊst sɪgˈnɪfɪkənt bɪt/ | The **most significant bit** is the leftmost bit. |
| **two's complement** /tuːz ˈkɒmplɪmənt/ | In **two's complement**, 11111111 means −1.<br>11111111 −1 |
| **signed integer** /saɪnd ˈɪntɪdʒə/ | A **signed integer** can be positive or negative. |
| **sign bit** /saɪn bɪt/ | A **sign bit** of 1 shows a negative number.<br>1 |
| **arithmetic shift** /ˌærɪθˈmetɪk ʃɪft/ | An **arithmetic shift** to the right halves the number and keeps its sign. |
| **one's complement** /wʌnz ˈkɒmplɪmənt/ | To get the **one's complement**, flip every bit. |
| **7-segment display** /ˈsevən ˈsegmənt dɪˈspleɪ/ | A digital clock shows each number on a **7-segment display**. |
| **memory address** /ˈmeməri əˈdres/ | Each byte in memory has its own **memory address**. |
| **code point** /kəʊd pɔɪnt/ | The letter A has the **code point** 65.<br>A 65 |
| **character set** /ˈkærɪktə set/ | ASCII is a **character set** for English letters.<br>ASCII |
| **encoding** /enˈkəʊdɪŋ/ | UTF-8 is a common **encoding** for text.<br>UTF-8 |
| **bitmap** /ˈbɪtmæp/ | A photo is stored as a **bitmap**. |
| **pixel** /ˈpɪksl/ | Each **pixel** is one small dot of colour. |
| **file header** /faɪl ˈhedə/ | The **file header** stores the image's width and height. |



exercises.lol exercises.com.cn

| Word | Example |
| --- | --- |
| **colour depth** /ˈkʌlə depθ/ | A **colour depth** of 8 bits gives 256 colours.<br>8 256 |
| **image resolution**<br>/ˈɪmɪdʒ ˌrezəˈluːʃn/ | An **image resolution** of 1920 × 1080 has about two million pixels.<br>1920 × 1080 |
| **screen resolution**<br>/skriːn ˌrezəˈluːʃn/ | A bigger **screen resolution** shows more detail. |
| **bit depth** /bɪt depθ/ | A higher **bit depth** gives more colours. |
| **frame** /freɪm/ | Each **frame** of the video is one picture. |
| **compression** /kəmˈpreʃn/ | **Compression** makes the file smaller. |
| **frame rate** /freɪm reɪt/ | A **frame rate** of 30 means 30 pictures every second.<br>30 30 |
| **vector graphic** /ˈvektə ˈgræfɪk/ | A **vector graphic** stays sharp when you make it bigger. |
| **drawing list** /ˈdrɔːɪŋ lɪst/ | The **drawing list** stores every shape in the picture. |
| **drawing objects** /ˈdrɔːɪŋ ˈɒbdʒekts/ | Circles and lines are **drawing objects**. |
| **primitive** /ˈprɪmɪtɪv/ | A circle is a basic shape, or **primitive**. |
| **properties** /ˈprɒpətiz/ | A circle's **properties** include its centre, radius and colour. |
| **render** /ˈrendə/ | The computer must **render** the shapes before you can see them. |
| **analogue data** /ˈænəlɒg ˈdeɪtə/ | Sound in the air is **analogue data**. |



exercises.lol exercises.com.cn

| Word | Example |
| --- | --- |
| **digital data** /ˈdɪdʒɪtl ˈdeɪtə/ | A computer stores sound as **digital data**. |
| **sampling** /ˈsæmplɪŋ/ | **Sampling** measures the sound many times every second. |
| **sampling rate** /ˈsæmplɪŋ reɪt/ | A **sampling rate** of 44 100 Hz is used for music.<br>44 100 Hz |
| **sampling resolution** /ˈsæmplɪŋ ˌrezəˈluːʃn/ | A higher **sampling resolution** stores each sample more exactly. |
| **amplitude** /ˈæmplɪtjuːd/ | Loud sounds have a large **amplitude**. |
| **sample resolution** /ˈsæmpl ˌrezəˈluːʃn/ | A **sample resolution** of 16 bits gives 65 536 levels.<br>16 65 536 |
| **quantisation** /ˌkwɒntaɪˈzeɪʃn/ | **Quantisation** rounds each sample to the nearest level. |
| **bandwidth** /ˈbændwɪdθ/ | A fast connection has a high **bandwidth**. |
| **lossless** /ˈlɒsləs/ | **Lossless** compression keeps every bit of the original file. |
| **lossy** /ˈlɒsi/ | **Lossy** compression removes some detail to save space. |
| **run-length encoding** /rʌn leŋθ enˈkəʊdɪŋ/ | **Run-length encoding** stores AAAAA as 5A.<br>AAAAA 5A |
| **dictionary methods** /ˈdɪkʃənəri ˈmeθədz/ | **Dictionary methods** replace common words with short codes. |
| **Huffman coding** /ˈhʌfmən ˈkəʊdɪŋ/ | **Huffman coding** gives short codes to common letters. |



exercises.lol exercises.com.cn

| Word | Example |
| --- | --- |
| **entropy** /ˈentrəpi/ | A file with high **entropy** is hard to compress. |
| **spatial** /ˈspeɪʃl/ | **Spatial** compression saves space within one picture. |
| **temporal** /ˈtempərəl/ | **Temporal** compression saves space between video frames. |



exercises.lol exercises.com.cn