**Computer Vocabulary S04**

|  |  |
| --- | --- |
| **development process** phr. 开发过程 | **decompose** v. 分解 |
| **annotation** n. 注释 | **alternatively** adv. 或者，另外 |
| **mathematical** adj. 数学的 | **embed** v. 嵌入 |
| **console pane** n. 控制台窗口 | **instruction set** phr. 指令集 |
| **string interpolation** phr. 字符串插值 | **skeleton** n. 骨架 |
| **asterisk** n. 星号 | **symmetry** n. 对称性 |
| **scenario** n. 情景，场景 | **physical memory** phr. 物理内存 |
| **pixel** n. 像素 | **virtual memory** phr. 虚拟内存 |
| **dependency** n. 依赖关系 | **Off-Screen Rendering** phr. 离屏渲染 |
| **scribble** v. 潦草地写，涂鸦 | **On-Screen Rendering** phr. 屏幕渲染 |
| **parsing** n. 解析 | **millisecond** n. 毫秒 |
| **fractional** adj. 分数的，部分的 | **list scrolling lag** phr. 列表卡顿 |
| **preferred** adj. 首选的 | **rounded corner effect** phr. 圆角效果 |
| **interoperability** n. 互操作性，可复用性 | **processor** n. 处理器 |
| **precision** n. 精度 | **contiguous** adj. 连续的，相邻的 |
| **literal** n. 字面值 | **static dispatch** phr. 静态分发 |
| **decimal** adj/n. 十进制的，小数的 | **dynamic dispatch** phr. 动态分发 |
| **hexadecimal** adj. 十六进制的 | **existential type** phr. 存在类型 |
| **truncate** v. 截断 | **existential container** phr. 存在容器 |
| **alias** n. 别名 | **generic specialization** phr. 泛型特化 |

▲ **development process** [dɪˈvɛləpmənt ˈprɑsɛs] phr. 开发过程

eg. The development process for this project involves multiple phases, from requirements gathering to deployment. 这个项目的开发过程涉及多个阶段，从需求收集到部署。

eg. Type safety helps you catch and fix errors as early as possible in the development process.

类型安全有助于您在开发过程中尽早发现并修复错误。

▲ **annotation** [ˌænəˈteɪʃn] n. 注释

eg. We can add annotations to our code to explain complex logic or provide context for future developers. 我们可以添加注释来解释复杂的逻辑或为将来的开发者提供上下文。

▲ **mathematical** [ˌmæθəˈmætɪkl] adj. 数学的

eg. Machine learning algorithms rely heavily on mathematical concepts, such as linear algebra and calculus. 机器学习算法严重依赖于数学概念，如线性代数和微积分。

▲ **console pane** [kənˈsoʊl peɪn] phr. 控制台窗口

eg. We can view debugging information in the console pane of our development environment.

我们可以在开发环境的控制台窗口中查看调试信息。

▲ **string interpolation** [strɪŋ ɪn ˌ ɪnˌtɜːrpəˈleɪʃn] phr. 字符串插值

eg. String interpolation allows us to embed variable values into a string literal for dynamic content generation. 字符串插值允许我们将变量值嵌入到字符串文字中，以进行动态内容生成。

▲ **asterisk** [ˈæstərɪsk] n. 星号

eg. In Python, we can use an asterisk to unpack a list or tuple into individual values for function arguments. 在Python中，我们可以使用星号将列表或元组拆包为函数参数的单个值。

▲ **scenario** [səˈnæriəʊ] n. 情景，场景

eg. The software developer created a test **scenario** to simulate a real-life situation and identify potential issues. 软件开发人员创建了一个测试情景来模拟实际情况并识别潜在问题。

▲ **pixel** [ˈpɪksl] n. 像素

eg. The resolution of the image is determined by the number of **pixels** in each row and column.

图像的分辨率由每行和每列的像素数确定。

▲ **dependency** [dɪˈpendənsi] n. 依赖关系

eg. In software development, the Dependency Injection design pattern is commonly used to manage **dependencies** between different components of an application.

在软件开发中，依赖注入设计模式通常用于管理应用程序中不同组件之间的依赖关系。

▲ **scribble** [ˈskrɪbl] v. 潦草地写，涂鸦

eg. I often **scribble** notes on paper or whiteboard during meetings to keep track of ideas.

我常常在会议期间在纸上或白板上草草地写下笔记，以追踪想法。

▲ **parsing** [ˈpɑrsɪŋ] n. 解析

eg. When developing a compiler, **parsing** is an essential step that involves analyzing the structure of source code to generate an abstract syntax tree. 在开发编译器时，解析是必不可少的一步，它涉及分析源代码的结构以生成抽象语法树。

▷ parse [pɑːrs] v. 解析

▲ **fractional** [ˈfrækʃənəl] adj. 分数的，小数的；很小的，微不足道的

eg. In computer programming, it is common to represent **fractional** numbers using floating-point data types such as float or double. 在计算机编程中，通常使用浮点数据类型（例如float或double）表示分数。

▲ **preferred** [prɪˈfɜrd] adj. 首选的

eg. The **preferred** language for Android app development is Java, although Kotlin is becoming increasingly popular. Android应用程序开发的首选语言是Java，尽管Kotlin越来越受欢迎。

▲ **i****nteroperability** [ˌɪntərˌɑpərəˈbɪləti] n. 互操作性，互用性，可复用性

eg. **Interoperability** is an important aspect of software development that refers to the ability of different systems or components to work together seamlessly. 互操作性是软件开发的一个重要方面，它指不同系统或组件之间无缝协同工作的能力。

▲ **precision** [prɪˈsɪʒən] n. 精度

eg. When dealing with numerical calculations, it is important to use data types that provide sufficient **precision** to avoid rounding errors. 在处理数值计算时，重要的是使用提供足够精度以避免舍入误差的数据类型。

▲ **literal** [ˈlɪtərəl] n. 字面值

eg. In programming, a **literal** refers to a fixed value that is written directly into source code, such as a number or a string. 在编程中，字面值是指直接写入源代码中的固定值，例如数字或字符串。

eg. If you assign a **literal** value of 42 to a new constant without saying what type it is, Swift infers that you want the constant to be an Int. 如果你把一个字面量42赋予了一个没有定义类型的新常量，Swift会推断出你想要这个常量为Int型。

▲ **decimal** [ˈdesɪml] adj/n. 十进制的，小数的；小数

eg. In mathematics and computer programming, **decimal** numbers are represented using the base-10 numbering system. 在数学和计算机编程中，十进制数字使用十进制数码系统表示。

eg. The decimal 0.61 stands for 61 hundredths. 小数0.61代表61%。

▲ **hexadecimal** [ˌhɛksəˈdɛsəməl] adj. 十六进制的

eg. In computer programming, hexadecimal notation is often used to represent binary data in a more human-readable form. 在计算机编程中，十六进制表示法经常用于以更易读的形式表示二进制数据。

▲ **truncate** [trʌŋˈkeɪt] v. 截断

eg. When converting a floating-point number to an integer, it is common to **truncate** the decimal part using functions like floor() or int(). 在将浮点数转换为整数时，通常使用floor()或int()等函数截断小数部分。

▲ **alias** [ˈeɪliəs] n. 别名

eg. In database management systems, **aliases** are commonly used to provide alternative names for tables or columns in queries. 在数据库管理系统中，别名通常用于为查询中的表或列提供替代名称。

▲ **decompose** [ˌdiːkəmˈpoʊz] v. 分解

eg. In mathematics, it is often useful to **decompose** a complex problem into simpler components that can be solved independently. 在数学中，将复杂问题分解为可以独立解决的简单组件通常是很有用的。

eg. In Swift, you can decompose a tuple’s contents into separate constants or variables.

在Swift中，您可以将元组的内容分解为单独的常量或变量。

▲ **alternatively** [ɔːlˈtɜrnətɪvli] adv. 或者，另外

eg. **Alternatively**, you can use a third-party library to add support for additional features in your application. 或者，您可以使用第三方库为您的应用程序添加额外的功能支持。

▲ **embed** [ɪmˈbed] v. 嵌入

eg. To display a YouTube video on a webpage, you can **embed** the video using an iframe element. 为了在网页上显示YouTube视频，您可以使用iframe元素嵌入视频。

▲ **instruction set** [ɪnˈstrʌkʃən sɛt] phr. 指令集

eg. The **instruction set** of a processor determines what operations the processor can perform.

处理器的指令集决定了处理器能执行哪些操作。

▲ **skeleton** [ˈskelɪtn] n. 骨架

eg. A **skeleton** of an application refers to the basic structure and functionality without any details or specific features. 应用程序的骨架指的是没有任何细节或具体功能的基本结构和功能。

▲ **symmetry** [ˈsɪmətri] n. 对称性

eg. In graphic design, **symmetry** is often used to create a sense of balance and harmony in a design. 在图形设计中，对称常用于创建设计中的平衡和和谐感。

▲ **physical memory** [ˈfɪzɪkəl ˈmɛməri] phr. 物理内存

eg. **Physical memory**, also known as RAM, is the hardware component that temporarily stores data for the processor to access. 物理内存，也称为RAM，是临时存储数据以供处理器访问的硬件组件。

▲ **virtual memory** [ˈvɜːrtʃuəl ˈmɛməri] phr. 虚拟内存

eg. **Virtual memory** is a technique used by operating systems to expand the amount of available memory by temporarily transferring data from RAM to the hard drive. 虚拟内存是操作系统用来通过将数据临时转移从RAM到硬盘来扩展可用内存量的技术。

▲ **Off-Screen Rendering** phr. 离屏渲染

eg. **Off-screen rendering** is a technique used by graphics processing units (GPUs) to render images or graphics that are not visible on the screen.

离屏渲染是由图形处理单元（GPU）用于渲染不在屏幕上可见的图像或图形的技术。

▲ **On-Screen Rendering** phr. 屏幕渲染

eg. **On-screen rendering** refers to the process of rendering images or graphics directly onto the screen. 屏幕渲染是指将图像或图形直接渲染到屏幕上的过程。

▲ **millisecond** [ˈmɪlɪˌsɛkənd] n. 毫秒

eg. In computing, tasks are often measured in **milliseconds**, with one millisecond equaling one thousandth of a second. 在计算机中，任务通常用毫秒来衡量，一毫秒等于一秒的千分之一。

▲ **list scrolling lag** [ˈskroʊlɪŋ læɡ] phr. 列表卡顿

eg. If a user experiences **list scrolling lag**, it means that there is a delay in the scrolling of a list on a device. 如果用户遇到列表卡顿，则表示设备上的列表滚动存在延迟。

▲ **rounded corner effect** phr. 圆角效果

eg. The **rounded corner effect** is a design feature that adds curved edges to rectangular shapes for a softer, more approachable appearance. 圆角效果是一种设计特征，它为矩形形状添加了弯曲的边缘，以获得更柔和、更平易近人的外观。

▲ **processor** [ˈprɑsɛsər] n. 处理器

eg. The **processor**, also known as the central processing unit (CPU), is the brain of a computer system that executes instructions and performs calculations. 处理器，也称为中央处理器（CPU），是计算机系统的大脑，执行指令并执行计算。

▲ **contiguous** [kənˈtɪɡjuəs] adj. 连续的，相邻的

eg. In computer memory, **contiguous** blocks of memory refer to segments of memory that are adjacent to each other and can be accessed in a single operation. 在计算机内存中，连续的内存块是指相邻的内存段，可以在单个操作中访问。

▲ **static dispatch** [ˈstætɪk dɪˈspætʃ] phr. 静态分发

eg. In static dispatch, the method to be called is determined at compile time based on the static type of the object. 在静态分发中，要调用的方法是根据对象的静态类型在编译时确定的。

▲ **dynamic dispatch** [daɪˈnæmɪk dɪˈspætʃ] phr. 动态分发

eg. In **dynamic dispatch**, the method to be called is determined at runtime based on the actual type of the object. 在动态分发中，要调用的方法是根据对象的实际类型在运行时确定的。

▲ **existential type** [ˌeɡzɪˈstenʃəl taɪp] phr. 存在类型

eg. An **existential type** is a type that abstracts over concrete types and hides their implementation details, while still providing a way to operate on the values of those types. 存在类型是一种抽象类型，它抽象出了具体类型并隐藏了它们的实现细节，同时仍然提供一种操作这些类型的值的方法。

▲ **existential container** [ˌeɡzɪˈstenʃəl kənˈteɪnər] phr. 存在容器

eg. An **existential container** is a type-erased container that can hold values of any type that conforms to a particular protocol or satisfies a certain set of requirements. 存在容器是一种类型擦除容器，可以容纳符合特定协议或满足一定要求的任何类型的值。

▲ **generic specialization** [dʒəˈnerɪk ˌspeʃələˈzeɪʃən] phr. 泛型特化

eg. **Generic specialization** is the process of generating specialized versions of generic code for specific types, which can result in improved performance and reduced memory usage.

泛型特化是为特定类型生成泛型代码的专用版本的过程，这可能会提高性能并减少内存使用。

▷ **specialization** n. 专业，专业化，专门化；（生物）特化作用