1. Time （时间）
Time 是你主要的全局计时信息源，你可以在任何需要操作时间的系统中访问该资源。Bevy 在每帧开始时更新时间。

2. 增量时间
最常见的用例是“增量时间”（delta time）——即前一帧更新和当前帧更新之间经过的时间。这告诉你游戏运行的速度，以便你可以调整诸如移动和动画之类的内容。这样一来，无论游戏的帧率如何，一切都可以顺畅地进行并以相同的速度运行。

```
fn move_blue_circle(mut blue_circle: Query<&mut Transform,
With<BlueCircle>>, time: Res<Time>) {
    let Ok(mut transform) = blue_circle.get_single_mut()
else {
        return;
    };
    transform.translation.x += 200. * time.delta_seconds();
}
```

3. 计时器和秒表
还有一些工具可以帮助你跟踪特定的间隔或计时：Timer 和 Stopwatch 。你可以创建许多此类实例，以跟踪你想要的任何内容。你可以在自己的组件或资源中使用它们。

计时器和秒表需要 tick。你需要有一些系统调用 .tick(delta) 才能使它们前进，否则它将处于非活动状态。增量应该来自 Time 资源。

```
fn tick_timers(mut timers: Query<&mut BigCircleTimer>, time:
Res<Time>) {
    for mut timer in &mut timers {
        timer.0.tick(time.delta());
    }
}

fn tick_stopwatches(mut stopwatches: Query<&mut
SmallCircleStopwatch>, time: Res<Time>) {
    for mut stopwatch in &mut stopwatches {
        stopwatch.0.tick(time.delta());
    }
}
```

4. Timer （计时器）
Timer 允许你检测特定时间间隔何时过去。计时器有设定的持续时间。计时器可以是重复或不重复的。

两种类型都可以手动“重置”（从头开始计算时间间隔）和暂停（即使你继续 tick 它们也不会前进）。

重复计时器在达到设定的持续时间后会自动重置。

使用 .finished() 来检测计时器是否已达到设定的持续时间。如果需要仅在达到持续时间的确切时刻进行检测，请使用 .just_finished()。

```
fn spawn_small_circle(
    mut commands: Commands,
    big_circle_timers: Query<(Entity, &BigCircleTimer)>,
    mut meshes: ResMut<Assets<Mesh>>,
    mut materials: ResMut<Assets<ColorMaterial>>,
) {
    for (entity, BigCircleTimer(timer)) in
&big_circle_timers {
        if timer.just_finished() {
            let small_circle_handle =
Mesh2dHandle(meshes.add(Circle::new(30.)));
            for index in 0..12 {
                let mut transform =
Transform::from_translation(Vec3::X * 160.);
                transform.translate_around(
                    Vec3::ZERO,

Quat::from_rotation_z(std::f32::consts::TAU * index as f32 /
12.),
                );
                let small_circle = commands
                    .spawn((
                        MaterialMesh2dBundle {
                            mesh:
small_circle_handle.clone(),
                            material:
materials.add(Color::Srgba(Srgba::interpolate(
                                &css::RED,
                                &css::BLUE,
                                index as f32 / 12.,
                            ))),
                            transform,
                            ..default()
                        },

SmallCircleStopwatch(Stopwatch::new()),
                        Name::new("Small Circle"),
                    ))
                    .id();

commands.entity(entity).add_child(small_circle);
            }
        }
    }
}
```

请注意，Bevy 的计时器与典型的现实生活中的计时器（倒计时至零）不同。Bevy 的计时器从零开始计时，并向设定的持续时间计时。它们基本上就像带有额外功能的秒表：最大持续时间和可选的自动重置功能。

5. Stopwatch （秒表）
Stopwatch 可以让你跟踪自某一点以来已经过去了多少时间。

它只会不断累积时间，你可以使用 .elapsed() / .elapsed_secs() 检查，你可以随时手动重置它。

```
fn despawn_small_circles(
    mut commands: Commands,
    small_circles: Query<(Entity, &SmallCircleStopwatch)>,
) {
    for (entity, SmallCircleStopwatch(stopwatch)) in
&small_circles {
        if stopwatch.elapsed_secs() > 3. {
            commands.entity(entity).despawn();
        }
    }
}
```