



CS Peach

Powered by [Cremencing.com](https://cremencing.com)
CH ABAP Peach

FREE CHEAT SHEET [CREMENCING.COM](https://cremencing.com)

Modern ABAP Survival Kit RAP, BTP, Joule & VS Code

**EXTRA
BONUS**



CS Peach
cremencing.com
2026 Edition

"You are not behind — you are one learning sprint away. This card is the exact path from classic ABAP to the modern stack, and what to learn first."

Classic ABAP — today

- **SE80 / SAP GUI** — the old workbench
- Classic reports, Dynpro, ALV lists
- Direct table access, FORM routines
- Code by hand, alone, in the system
- On-premise only, tied to one DB
- Modifications — upgrade pain every time



Modern ABAP — 2026

- **Eclipse ADT & VS Code** — real tooling
- **RAP + Fiori** — full-stack apps, not lists
- Released APIs & **CDS** — clean core
- **Joule & AI** assist — you review, it drafts
- On-prem, private & public cloud — one model
- Extensions on top — upgrades stay cheap

Eclipse → VS Code

modern editor in 2 steps

RAP vs Classic Report

when to choose which

SAP Joule & ABAP

what it can & can't do

- 1 Install the **ABAP extension (ABAP Remote FS / abap-adt-api based)** in VS Code.
- 2 Connect to your system's **ADT service** — same backend Eclipse uses. Now edit ABAP in VS Code.

Choose RAP when: you need a real transactional app — create/update, Fiori UI, OData, draft handling, validations.

Choose a classic report when: it is a quick internal read-only list with no UI ambition and no cloud future.

The honest truth: almost all new user-facing work is RAP now.
CDS → behavior definition → service → Fiori. Learn this flow.

CAN: draft boilerplate, CDS & RAP skeletons, unit tests, explain code, suggest fixes, speed up the first 70%.

CAN'T: know your system's data, transports, naming, or edge cases — or take responsibility for what activates.

The pro move: let AI draft, **you review, test, and own it.** The developer who directs the AI stays essential; the one who blindly pastes does not.

Reality check: Eclipse ADT is still the most complete. VS Code is rising fast and pairs well with AI copilots — learn both, lead with Eclipse.

SKILLS TO ADD IN 2026 — IN PRIORITY ORDER

If you learn them top to bottom, each one makes the next easier.



Laeq Siddique

Free resource by Cremencing Solutions cremencing.com
CS Peach-Alconsultant for SAP ABAP teams



CREMENCING
SOLUTIONS



FREE CHEAT SHEET CREMENCING.COM

Free BTP Setup, Your First RAP App & the AI Shift

EXTRA
BONUS

CSPeach
cremencing.com
2026 Edition

"Reading about modern ABAP won't move you. Building one RAP app on a free BTP account will. Here is the exact path — free."

Start BTP for Free — today

no company account needed

- 1 Sign up for the **SAP BTP Free Tier** (or the trial) at the SAP BTP site — personal email works.
- 2 Create a **subaccount** and enable the **ABAP environment** (the "steampunk" ABAP Cloud).
- 3 Open the system in **Eclipse ADT** with the ABAP Development Tools plugin — connect and log in.
- 4 You now have a **real ABAP Cloud system** to build on — free, yours, safe to experiment in.

Other free routes: the **ABAP Platform Developer Edition** (Docker image, full on-prem ABAP), SAP Learning tutorials with built-in systems, and the SAP Developer Center trials.

"You do not need your employer's system to learn this. A free BTP account and one weekend is enough to build your first RAP app."

Build Your First RAP App

the flow every modern ABAPer knows

- 1 Create a **database table** (or reuse one) as your data foundation.
- 2 Build a **CDS view** on it — this exposes and shapes the data.
- 3 Add a **behavior definition** — create, update, delete, validations, draft.
- 4 Expose it as an **OData service** via a service definition + binding.
- 5 Preview the **Fiori Elements app** — a working UI with no JavaScript written.

That five-step flow — table → CDS → behavior → service → Fiori — is the heart of modern ABAP. Do it once and it clicks forever.

Why the Best Are Adopting AI

quietly, and on purpose

It removes the boring 70%

Boilerplate, CDS scaffolding, test stubs, documentation — AI drafts them in seconds so you spend your time on the hard 30%.

It is a teacher, not just a tool

Ask it to **explain** unfamiliar RAP or CDS syntax. Many devs learn the modern stack faster *with* AI than from docs alone.

The risk if you ignore it

The gap between AI-fluent and AI-avoiding developers is widening fast. In two years it will be the main line between senior and stuck.

The safe way to adopt

Review everything, test everything, own everything. AI drafts; you are still the engineer who signs off. That is exactly what CSPeach is built around.

YOUR 30-DAY MODERNIZATION SPRINT

- **Week 1:** free BTP account + Eclipse ADT connected. Read + write your first CDS view.
- **Week 2:** build the 5-step RAP app end to end. See your own Fiori UI.
- **Week 3:** add validations, a determination, and an ABAP Unit test.
- **Week 4:** use AI to draft, then review & harden it. Post what you built.

SAY THIS IN A MODERN-ABAP INTERVIEW

- **"Do you know RAP?"** — "Yes — table, CDS, behavior definition, service, Fiori Elements. I've built the full flow."
- **"How do you use AI in ABAP?"** — "To draft and explain — then I review, test, and own the result. It never activates unchecked."
- **"Eclipse or VS Code?"** — "Eclipse ADT for depth, VS Code for AI-assisted flow. I work in both."

THE ONE PRINCIPLE THAT TIES IT TOGETHER

- **You don't need permission or a company system to become a modern ABAP developer.**
- A free BTP account, one RAP app, and AI as your co-pilot — that is the whole on-ramp.
- The developers who feel "left behind" are one focused month from being ahead.
- **Direct the tools, own the output — and you stay irreplaceable in the AI era.**



Laeq Siddique

Free resource by Cremencing Solutions cremencing.com
CSPeach-AIconsultant for SAP ABAP teams

 **CREMENCING**
SOLUTIONS